# SHRI VASIHNAV VIDHYAPEETH VISHWAVIDHYALAYA, INDORE


TOPIC: Face recognition based attendance system

SUBJECT: Scripting Language

SUBMITTED TO:
Prof.Avdhesh Kumar Sharma

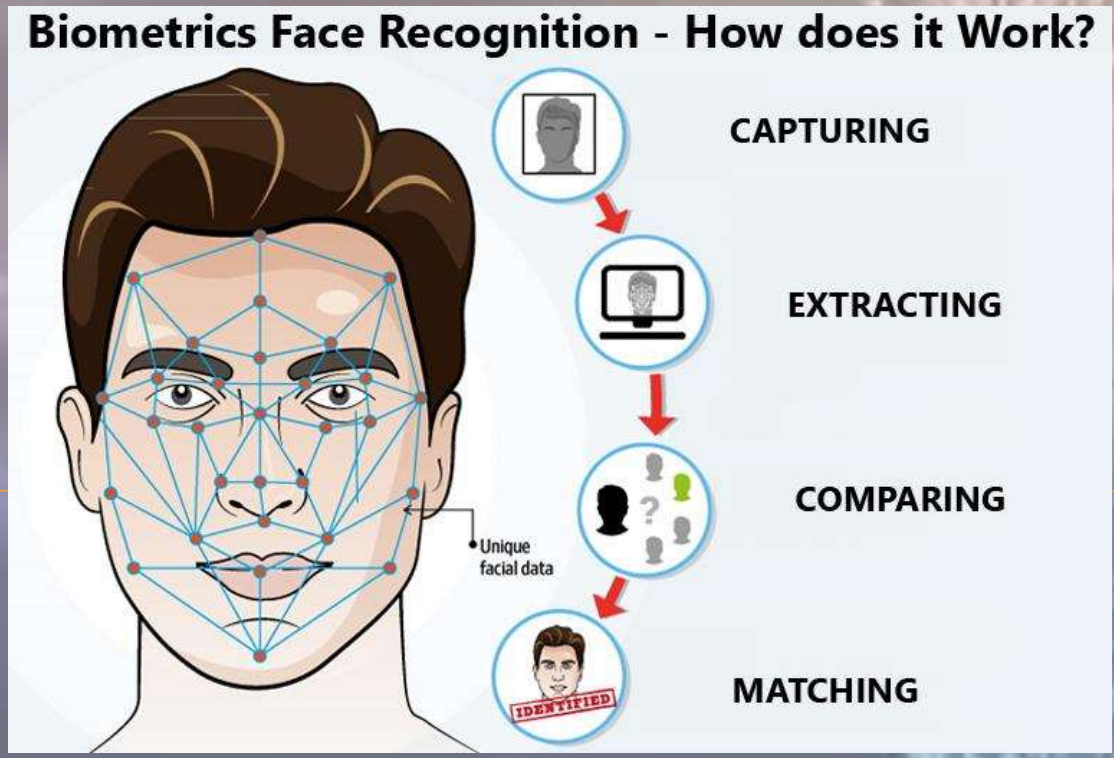
SUBMITTED BY:
Harshita Kumawat: 20100BTCSE07564
Isha Choudhary: 20100BTCSE07569
Kanchan Giri: 20100BTCSE07576

# CONTENTS:



# Abstract

A facial recognition systaem is a technology  capable of matching a human face from a digital image or a video frame against a database of faces, typically employed to authenticate users through ID verification services, works by pinpointing and measuring facial features from a given image



# Problem statement

- Attendance is very important part of daily classroom evaluation.
- At the beginning and ending of the class it is usually checked by the teacher , but it may occur that a teacher may miss someone or some students answer multiple times
- Face recognition based attendance system is a problem of recognizing face for taking attendance by using face recognition technology based on high definition monitor video and other information technology

# What Is face recognition?

- Facial recognition is a way of identifying or confirming an individual's identity using their face. Facial recognition systems can be used to identify people in photos, videos, or in real-time.
- Facial recognition is a category of biometric security. Other forms of biometric software include voice recognition, fingerprint recognition, and eye retina or iris recognition. The technology is mostly used for security and law enforcement, though there is increasing interest in other areas of use.

# Benefits of face recognition?

- Efficient security
- Improved accuracy
- Easier integration

# What is scripting language?

- A scripting language is a programming language that employs a high-level construct to interpret and execute one command at a time
- Compiled languages are converted permanently into executable_files before they are run. In contrast, scripting languages are typically converted into machine code on the fly during runtime by a program called an interpreter.
- In this project we used php as scripting language
- And html and css for frontend

# What is php?

- PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

# What is opencv?

- 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.
- It mainly focuses on image processing, video capture and analysis including features like face detection and object detection.

# What Is numpy?

- NumPy stands for Numerical Python
- s a library consisting of multidimensional array objects and a collection of routines for processing those arrays
- Using NumPy, mathematical and logical operations on arrays can be performed

# Source code:

```
import tkinter as tk from tkinter
 import *
import cv2
 import csv
import os
import numpy as np from PIL
 import Image
,ImageTk import pandas as pd
import datetime
import time
 #####Window is our Main frame of system window = tk.Tk()
window.title("FaceReqG-Face Recognition Based Attendance Management
System")
window.geometry('1280x720') window.configure(background='azure2')

####GUI for manually fill attendance def manually_fill(): global sb sb = tk.Tk()
    sb.title("Enter subject name...") sb.geometry('580x320')
    sb.configure(background='azure2')
```

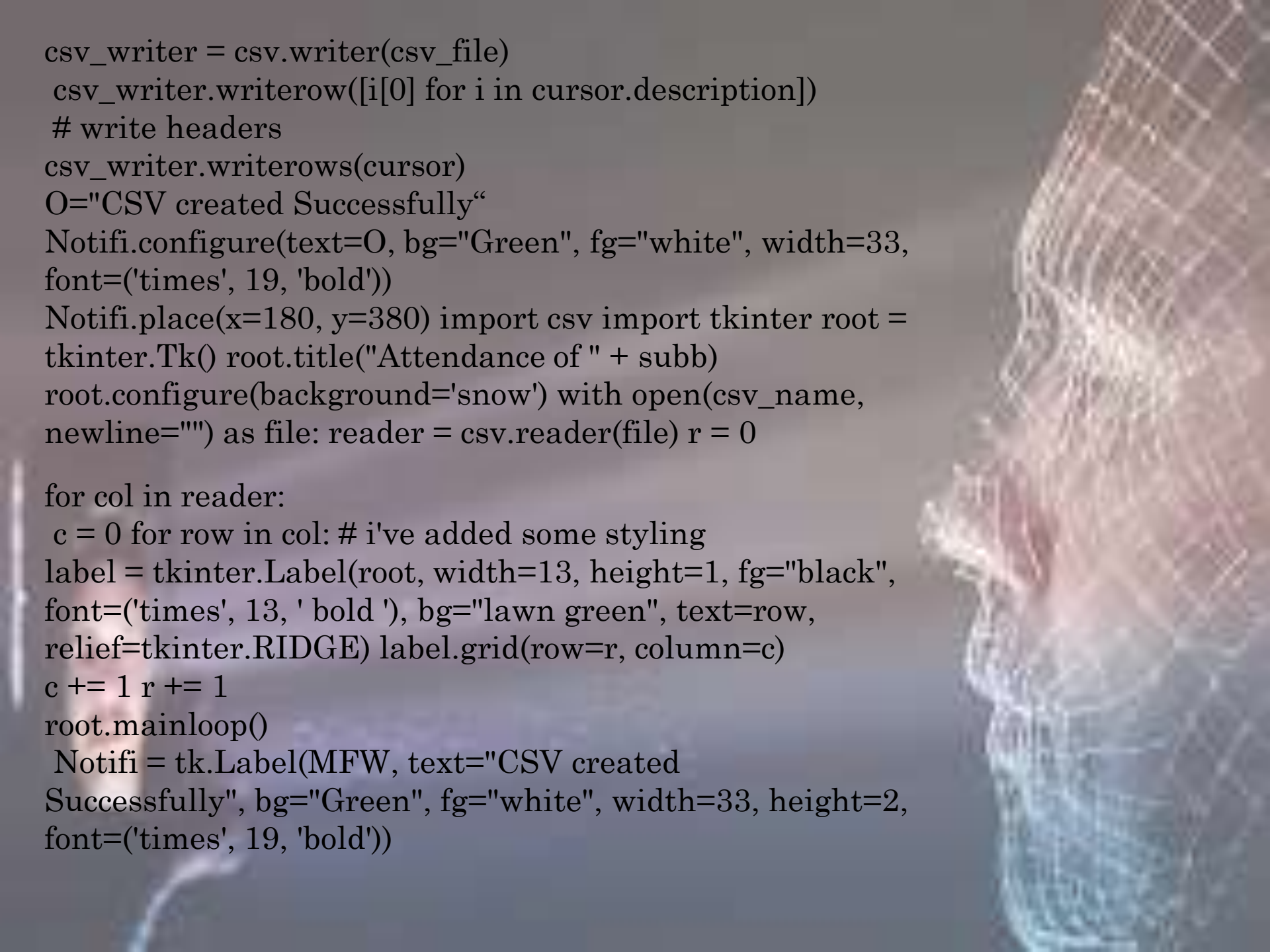
```
def err_screen_for_subject():
    def ec_delete():
        ec.destroy()
        global ec ec = tk.Tk()
        ec.geometry('300x100')
        ec.title('Warning!!')
        ec.configure(background='burlywood')
        Label(ec, text='Please enter your subject name!!!', fg='red',
bg='white', font=('times', 16, ' bold ')).pack() Button(ec, text='OK',
command=ec_delete, fg="black", bg="lawn green", width=9, height=1,
activebackground="Red", font=('times', 15, ' bold ')).place(x=90, y=50)
def fill_attendance():
ts = time.time()
Date = datetime.datetime.fromtimestamp(ts).strftime('%Y_%m_%d')
timeStamp =
datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S') Time =
datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S') Hour,
Minute, Second = timeStamp.split(":") ####Creating csv of
attendance
```

- ##Create table for Attendance date_for_DB = datetime.datetime.fromtimestamp(ts).strftime('%Y_%m_%d') global subb subb=SUB_ENTRY.get() DB_table_name = str(subb + "_" + Date + "_Time_" + Hour + "_" + Minute + "_" + Second)

```
import py mysql.connections
###Connect to the database
 try:
 global cursor connection = pymysql.connect(host='localhost', user='root', password='',
db='manually_fill_attendance')
cursor = connection.cursor() except Exception as e: print(e) sql = "CREATE TABLE " +
DB_table_name + """ (ID INT NOT NULL AUTO_INCREMENT, ENROLLMENT varchar(100)
NOT NULL, NAME VARCHAR(50) NOT NULL, DATE VARCHAR(20) NOT NULL, TIME
VARCHAR(20) NOT NULL, PRIMARY KEY (ID) ); """
try: cursor.execute(sql)
 #for
create a table connection.commit() except Exception as ex:
 print(ex)
 #
f subb==': err_screen_for_subject()
 else: sb.destroy() MFW = tk.Tk() MFW.title("Manually attendance of "+ str(subb))
MFW.geometry('880x470') MFW.configure(background='snow')
def del_errsc2():
 errsc2.destroy()
 def err_screen1():
 global errsc2
errsc2 = tk.Tk())
```

```
errsc2.geometry('330x100') errsc2.title('Warning!!') errsc2.configure(background='snow')
 Label(errsc2, text='Please enter Student & Enrollment!!!', fg='red', bg='white',
      font=('times', 16, ' bold ')).pack()
Button(errsc2, text='OK', command=del_errsc2, fg="black", bg="lawn green", width=9,
      height=1,
activebackground="Red", font=('times', 15, ' bold ')).place(x=90, y=50) def testVal(inStr,
      acttyp):
 if acttyp == '1': # insert if not inStr.isdigit(): return False return True ENR =
      tk.Label(MFW, text="Enter
Enrollment", width=15, height=2, fg="white", bg="blue2", font=('times', 15, ' bold '))
ENR.place(x=30, y=100) STU_NAME = tk.Label(MFW, text="Enter Student name",
      width=15, height=2,
fg="white", bg="blue2", font=('times', 15, ' bold ')) STU_NAME.place(x=30, y=200) global
      ENR_ENTRY
ENR_ENTRY = tk.Entry(MFW, width=20,validate='key', bg="yellow", fg="red",
      font=('times', 23, ' bold '))
ENR_ENTRY['validatecommand'] = (ENR_ENTRY.register(testVal), '%P', '%d')
      ENR_ENTRY.place(x=290,
y=105) def remove_enr(): ENR_ENTRY.delete(first=0, last=22
STUDENT_ENTRY = tk.Entry(MFW, width=20, bg="yellow", fg="red",
font=('times', 23, ' bold ')) STUDENT_ENTRY.place(x=290, y=205)
def remove_student(): STUDENT_ENTRY.delete(first=0, last=22)
```

```
####get important variable
 def enter_data_DB():
ENROLLMENT = ENR_ENTRY.get()
STUDENT = STUDENT_ENTRY.get()
 if ENROLLMENT==":
err_screen1()
 elif STUDENT==":
 err_screen1() else:
time = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
Hour, Minute, Second = time.split(":")
 Insert_data = "INSERT INTO " + DB_table_name + "
(ID,ENROLLMENT,NAME,DATE,TIME) VALUES (0, %s, %s, %s,%s)"
VALUES = (str(ENROLLMENT), str(STUDENT), str(Date), str(time))
try: cursor.execute(Insert_data, VALUES) connection.commit() except
Exception as e: print(e) ENR_ENTRY.delete(first=0, last=22)
STUDENT_ENTRY.delete(first=0, last=22)
def create_csv():
 import csv cursor.execute("select * from " + DB_table_name
+ ";") connection.commit()
csv_name='StudentDetailcreate'+DB_table_name+'.csv' with
open(csv_name, "w") as csv_file:
```



```
csv_writer = csv.writer(csv_file)
 csv_writer.writerow([i[0] for i in cursor.description])
 # write headers
csv_writer.writerows(cursor)
O="CSV created Successfully“
Notifi.configure(text=O, bg="Green", fg="white", width=33,
font=('times', 19, 'bold'))
Notifi.place(x=180, y=380) import csv import tkinter root =
tkinter.Tk() root.title("Attendance of " + subb)
root.configure(background='snow') with open(csv_name,
newline="") as file: reader = csv.reader(file) r = 0

for col in reader:
 c = 0 for row in col: # i've added some styling
label = tkinter.Label(root, width=13, height=1, fg="black",
font=('times', 13, ' bold '), bg="lawn green", text=row,
relief=tkinter.RIDGE) label.grid(row=r, column=c)
c += 1 r += 1
root.mainloop()
 Notifi = tk.Label(MFW, text="CSV created
Successfully", bg="Green", fg="white", width=33, height=2,
font=('times', 19, 'bold'))
```

```
c1ear_enroll = tk.Button(MFW,
text="Clear", command=remove_enr, fg="black", bg="deep pink",
width=10, height=1, activebackground="Red", font=('times', 15, '
bold ')) c1ear_enroll.place(x=690, y=100)
 c1ear_student =
tk.Button(MFW, text="Clear", command=remove_student,
fg="black", bg="deep pink", width=10, height=1,
activebackground="Red", font=('times', 15, ' bold '))
c1ear_student.place(x=690, y=200)

DATA_SUB = tk.Button(MFW, text="Enter
Data",command=enter_data_DB, fg="black", bg="lime green", width=20,
height=2, activebackground="Red", font=('times', 15, ' bold '))
DATA_SUB.place(x=170, y=300) MAKE_CSV = tk.Button(MFW,
text="Convert to CSV",command=create_csv, fg="black", bg="red",
width=20, height=2, activebackground="Red", font=('times', 15, ' bold '))
MAKE_CSV.place(x=570, y=300) def attf():
```

```
import subprocess
subprocess.Popen(r'explorer /select,"StudentDetailentercheckatt.csv"')
attf = tk.Button(MFW, text="Check Sheets",command=attf,fg="black"
,bg="lawn green" ,width=12 ,height=1 ,activebackground = "Red"
,font=('times', 14, ' bold ')) attf.place(x=730, y=410) MFW.mainloop()
SUB = tk.Label(sb, text="Enter Subject", width=15, height=2,
fg="white", bg="blue2", font=('times', 15, ' bold ')) SUB.place(x=30,
y=100)

global SUB_ENTRY SUB_ENTRY = tk.Entry(sb, width=20, bg="yellow", fg="red",
font=('times', 23, ' bold ')) SUB_ENTRY.place(x=250, y=105) fill_manual_attendance =
tk.Button(sb, text="Fill Attendance",command=fill_attendance, fg="white", bg="deep
pink", width=20, height=2, activebackground="Red", font=('times', 15, ' bold ')) f
fill_manual_attendance.place(x=250, y=160)
sb.mainloop()
 ##For clear textbox
def clear():
 txt.delete(first=0, last=22) def clear1():
txt2.delete(first=0, last=22) def del_sc1(): sc1.destroy() def err_screen(): global sc1 sc1 =
tk.Tk() sc1.geometry('300x100') sc1.title('Warning!!') sc1.configure(background='snow')
Label(sc1,text='Enrollment & Name required!!!',fg='red',bg='white',font=('times', 16, ' bold
')).pack() Button(sc1,text='OK',command=del_sc1,fg="black" ,bg="lawn green" ,width=9
```

```
##Error screen2
 def del_sc2():
 sc2.destroy()
def err_screen1():
 global sc2
sc2 = tk.Tk()
 sc2.geometry('300x100') sc2.title('Warning!!')
sc2.configure(background='snow')
Label(sc2,text='Please enter your subject
name!!!',fg='red',bg='white',font=('times', 16, ' bold ')).pack()
Button(sc2,text='OK',command=del_sc2,fg="black" ,bg="lawn green"
,width=9 ,height=1, activebackground = "Red" ,font=('times', 15, ' bold
')).place(x=90,y= 50)
 ###For take images for datasets def take_img(): l1
= txt.get()
l2 = txt2.get()
if l1 == '': err_screen() elif l2 == '': err_screen()
else: try: cam = cv2.Vid
eoCapture(0) detector =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
Enrollment = txt.get()
Name = txt2.get() sampleNum = 0 while (True):
ret, img = cam.read()
gray = cv2.cvtColor(img,
cv2.COLOR_BGR2GRAY) faces = detector.detectMultiScale(gray, 1.3, 5)
for (x, y, w, h) in faces: cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
```

```
# incrementing sample number
sampleNum = sampleNum + 1
 # saving the captured face in the dataset folder
cv2.imwrite("ImageTrain/ " + Name + "." + Enrollment + '.' + str(sampleNum) + ".jpg", gray[y:y + h, x:x + w])
cv2.imshow('Frame', img)
 # wait for 100 miliseconds
if cv2.waitKey(1) & 0xFF == ord('q'): break # break if the sample number is morethan 100 elif sampleNum 70:
break
cam.release()
cv2.destroyAllWindows()
ts = time.time() Date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d') Time = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S') row = [Enrollment, Name, Date, Time] with open('StudentDetailtest.csv', 'a+') as csvFile: writer = csv.writer(csvFile, delimiter=',') writer.writerow(row) csvFile.close() res = "Images Saved for Enrollment : " + Enrollment + " Name : " + Name Notification.configure(text=res, bg="SpringGreen3", width=50, font=('times', 18, 'bold')) Notification.place(x=250, y=400) except FileExistsError as F: f = 'Student Data already exists' Notification.configure(text=f, bg="Red", width=21)
Notification.place(x=450, y=400)
```

```
###for choose subject and fill attendance
 def subjectchoose():
 def Fillattendances():
sub=tx.get()
 now = time.time()
###For calculate seconds of video
 future = now + 20 if time.time() < future: if sub == '': err_screen1() else:
     recognizer = cv2.face.LBPHFaceRecognizer_create()
 # cv2.createLBPHFaceRecognizer()
 #taking image
 try:
recognizer.read("TrainingImage")
except: e = 'Model not found,Please
train model' Notifica.configure(text=e, bg="red", fg="black", width=33,
font=('times', 15, 'bold')) Notifica.place(x=20, y=250) harcascadePath =
"haarcascade_frontalface_default.xml" faceCascade =
cv2.CascadeClassifier(harcascadePath) df =
pd.read_csv("StudentDetailmanatt.csv") cam = cv2.VideoCapture(0) font
= cv2.FONT_HERSHEY_SIMPLEX col_names = ['Enrollment', 'Name',
'Date', 'Time'] attendance = pd.DataFrame(columns=col_names)
```

```
while True:
 ret, im = cam.read()
gray = cv2.cvtColor(im,
cv2.COLOR_BGR2GRAY)
faces = faceCascade.detectMultiScale(gray, 1.2, 5) for
(x, y, w, h) in faces: global Id Id, conf = recognizer.predict(gray[y:y + h, x:x + w])
if (conf <70): print(conf) global Subject global aa global date global timeStamp
Subject = tx.get() ts = time.time() date =
    datetime.datetime.fromtimestamp(ts).strftime('%Y-
%m-%d') timeStamp =
    datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
 aa = df.loc[df['Enrollment'] == Id]['Name'].values global tt tt = str(Id) + "-" + aa
    En =
'15624031' + str(Id) attendance.loc[len(attendance)] = [Id, aa, date, timeStamp]
cv2.rectangle(im, (x, y), (x + w, y + h), (0, 260, 0), 7) cv2.putText(im, str(tt), (x +
h, y), font, 1, (255, 255, 0,), 4) else: Id = 'Unknown' tt = str(Id) cv2.rectangle(im,
(x, y), (x + w, y + h), (0, 25, 255), 7) cv2.putText(im, str(tt), (x + h, y), font, 1, (0,
25, 255), 4) if time.time() > future: break
```

```
attendance = attendance.drop_duplicates(['Enrollment'], keep='first')
cv2.imshow('Filling attedance..', im)
 key = cv2.waitKey(30) & 0xff if key == 27: break ts = time.time()
date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
 timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S') Hour, Minute,
Second = timeStamp.split(":") fileName = "Attendance/" + Subject + "_" + date + "_" + Hour + "-"
+ Minute + "-" + Second + ".csv" attendance = attendance.drop_duplicates(['Enrollment'],
keep='first')
print(attendance) attendance.to_csv(fileName, index=False)
##Create table for Attendance
date_for_DB = datetime.datetime.fromtimestamp(ts).strftime('%Y_%m_%d')
DB_Table_name = str( Subject + "_" + date_for_DB + "_Time_" + Hour + "_" + Minute + "_" +
      Second)
import
pymysql.connections
 ###Connect to the database try: global cursor connection
pymysql.connect(host='localhost', user='root', password='', db='Face_reco_fill') cursor =
connection.cursor() except Exception as e: print(e) sql = "CREATE TABLE " + DB_Table_name +
""" (ID INT NOT NULL AUTO_INCREMENT, ENROLLMENT varchar(100) NOT NULL,
NAME VARCHAR(50) NOT NULL, DATE VARCHAR(20) NOT NULL, TIME VARCHAR(20)
NOT NULL, PRIMARY KEY (ID) ); """ ####Now enter attendance in Database insert_data =
"INSERT INTO " + DB_Table_name + " (ID,ENROLLMENT,NAME,DATE,TIME) VALUES (0,
%s, %s, %s,%s)"
```

```
VALUES = (str(Id), str(aa), str(date), str(timeStamp))
 try: cursor.execute(sql)
##for create a table connection.commit()
cursor.execute(insert_data, VALUES)
##For insert data into table
connection.commit()
except Exception as ex: print(ex)
 # M = 'Attendance
filled Successfully' Notifica.configure(text=M, bg="Green", fg="white",
width=33, font=('times', 15, 'bold')) Notifica.place(x=20, y=250)
cam.release()
 cv2.destroyAllWindows()
import csv import tkinter root = tkinter.Tk() root.title("Attendance of " +
Subject) root.configure(background='snow') cs = " + fileName with open(cs,
newline="") as file: reader = csv.reader(file) r = 0 for col in reader: c = 0
for row in col: # i've added some styling label = tkinter.Label(root,
width=8, height=1, fg="black", font=('times', 15, ' bold '), bg="lawn
green", text=row, relief=tkinter.RIDGE) label.grid(row=r, column=c) c
+= 1 r += 1 root.mainloop() print(attendance)
```

```
###windo is frame for subject chooser
windo = tk.Tk()
windo.title("Enter subject name...")
windo.geometry('580x320')
windo.configure(background='snow')
 Notifica = tk.Label(windo, text="Attendance filled Successfully", bg="Green",
fg="white", width=33, height=2, font=('times', 15, 'bold'))
def Attf():
 import subprocess subprocess.Popen(r'explorer
select,"StudentDetaileteircheckatt.csv"') attf = tk.Button(windo, text="Check
Sheets",command=Attf,fg="black" ,bg="lawn green" ,width=12 ,height=1
,activebackground = "Red" ,font=('times', 14, ' bold ')) attf.place(x=430, y=255)
sub = tk.Label(windo, text="Enter Subject", width=15, height=2, fg="white",
bg="blue2", font=('times', 15, ' bold ')) sub.place(x=30, y=100) tx =
tk.Entry(windo, width=20, bg="yellow", fg="red", font=('times', 23, ' bold '))
tx.place(x=250, y=105) fill_a = tk.Button(windo, text="Fill Attendance",
fg="white",command=Fillattendances, bg="deep pink", width=20, height=2,
activebackground="Red", font=('times', 15, ' bold ')) fill_a.place(x=250, y=160)
windo.mainloop()
```

```
def admin_panel():
 win = tk.Tk() win.title("LogIn")
win.geometry('880x420') win.configure(background='snow')
 def log_in():
username = un_entr.get()
 password = pw_entr.get() if username == 'FaceReqG' :
#password and userid of an app if password == 'FaceReqG':
win.destroy()
import csv import tkinter
root = tkinter.Tk()
root.title("Student Details")
root.configure(background='snow')
 cs = 'StudentDetailreg.csv'
with open(cs, newline="") as file: reader = csv.reader(file) r = 0 for
col in reader: c = 0 for row in col: # i've added some styling label =
tkinter.Label(root, width=8, height=1, fg="black", font=('times',
15, ' bold '), bg="lawn green", text=row, relief=tkinter.RIDGE)
label.grid(row=r, column=c) c += 1 r += 1 root.mainloop()
```

```
else: valid = 'Incorrect ID or Password' Nt.configure(text=valid, bg="red", fg="black", width=38,
font=('times', 19, 'bold')) Nt.place(x=120, y=350) else: valid ='Incorrect ID or Password'
Nt.configure(text=valid, bg="red", fg="black", width=38, font=('times', 19, 'bold'))
Nt.place(x=120, y=350) Nt = tk.Label(win, text="Attendance filled Successfully", bg="Green",

fg="white", width=40, height=2, font=('times', 19, 'bold')) # Nt.place(x=120, y=350) un =
tk.Label(win, text="Enter username", width=15, height=2, fg="white", bg="blue2", font=('times',
15, ' bold ')) un.place(x=30, y=50) pw = tk.Label(win, text="Enter password", width=15,
height=2, fg="white", bg="blue2", font=('times', 15, ' bold ')) pw.place(x=30, y=150) def c00():
un_entr.delete(first=0, last=22) un_entr = tk.Entry(win, width=20, bg="yellow", fg="red",
font=('times', 23, ' bold ')) un_entr.place(x=290, y=55) def c11(): pw_entr.delete(first=0, last=22)
pw_entr = tk.Entry(win, width=20,show="*", bg="yellow", fg="red", font=('times', 23, ' bold '))
pw_entr.place(x=290, y=155) c0 = tk.Button(win, text="Clear", command=c00, fg="black",
bg="deep pink", width=10, height=1, activebackground="Red", font=('times', 15, ' bold '))
c0.place(x=690, y=55) c1 = tk.Button(win, text="Clear", command=c11, fg="black", bg="deep
pink", width=10, height=1, activebackground="Red", font=('times', 15, ' bold ')) c1.place(x=690,

y=155) Login = tk.Button(win, text="LogIn", fg="black", bg="lime green", width=20, height=2,
activebackground="Red",command=log_in, font=('times', 15, ' bold ')) Login.place(x=290, y=250)
    win.mainloop()
```

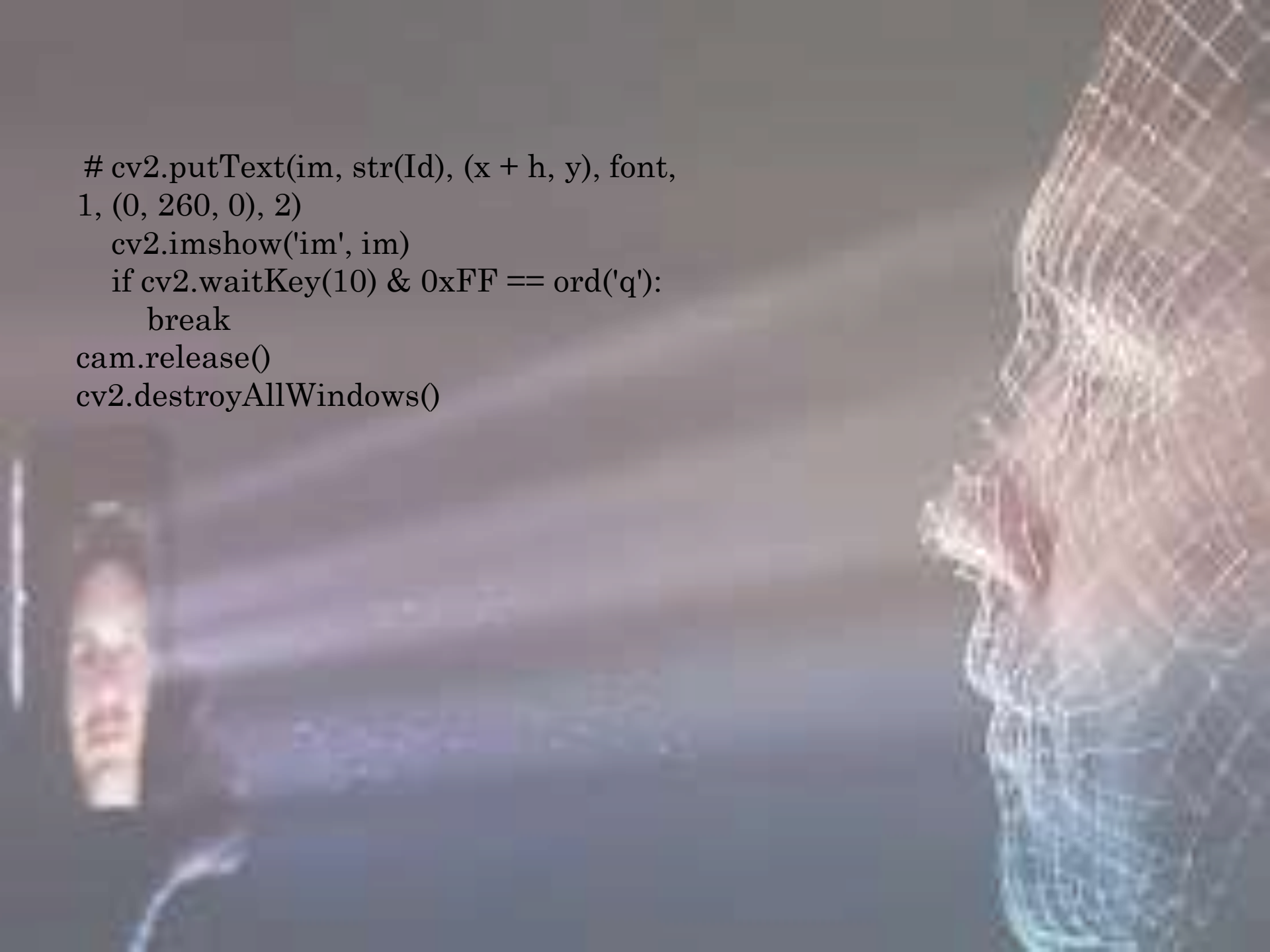
```
###For train the model def trainimg():
 recognizer = cv2.face.LBPHFaceRecognizer_create()
global detector detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
 try:
global faces,Id faces, Id = getImagesAndLabels("TrainingImage") except Exception as e:
l='please make "TrainingImage" folder & put Images' Notification.configure(text=l,
bg="SpringGreen3", width=50, font=('times', 18, 'bold')) Notification.place(x=350, y=400) recognizer.train(faces,
np.array(Id))
 try: recognizer.save("TrainingImage")
 except Exception as e: q='Please make "TrainingImage" folder' Notification.configure(text=q,
bg="SpringGreen3", width=50, font=('times', 18, 'bold')) Notification.place(x=350, y=400) res = "Model Trained"
# +",".join(str(f) for f in Id) Notification.configure(text=res, bg="SpringGreen3", width=50,
font=('times', 18, 'bold')) Notification.place(x=250, y=400) def getImagesAndLabels(path): imagePaths
[os.path.join(path, f) for f in os.listdir(path)]
 # create empth face list
faceSamples = []
# create empty ID
 list Ids = []
 # now looping through all the image paths and loading the Ids and the images for imagePath in imagePaths: #
loading the image and converting it to gray scale pilImage = Image.open(imagePath).convert('L')
 # Now we are converting the PIL image into numpy array imageNp = np.array(pilImage, 'uint8')
# getting the Id from the image Id = int(os.path.split(imagePath)[-1].split(".")[1]) # extract the face from the
training image sample faces =
detector.detectMultiScale(imageNp) # If a face is there then append that in the list as well as Id of it for (x, y,
w, h) in faces: faceSamples.append(imageNp[y:y + h, x:x + w]) Ids.append(Id) return faceSamples, Ids
window.grid_rowconfigure(0, weight=1) window.grid_columnconfigure(0, weight=1)
```

```
def on_closing():
from tkinter
 import messagebox if messagebox.askokcancel("Quit", "Do you want to quit?"):
 window.destroy()
window.protocol("WM_DELETE_WINDOW", on_closing) message = tk.Label(window,
text="Face-Recognition-Based-Attendance-Management-System", bg="cyan", fg="black",
width=50, height=3, font=('times', 30, 'italic bold '))
 message.place(x=80, =20) Notification = tk.Label(window, text="All things good", bg="Green",
fg="white", width=15, height=3, font=('times', 17, 'bold')) lbl = tk.Label(window, text="Enter
Enrollment", width=20, height=2, fg="black", bg="deep pink", font=('times', 15, ' bold '))
lbl.place(x=200, y=200)
def testVal(inStr,acttyp):
 if acttyp == '1':
 #insert if not inStr.isdigit():
 return False return True txt = tk.Entry(window, validate="key", width=20, bg="yellow",
fg="red", font=('times', 25, ' bold ')) txt['validatecommand'] = (txt.register(testVal),'%P','%d')
txt.place(x=550, y=210) lbl2 = tk.Label(window, text="Enter Name", width=20, fg="black",
bg="deep pink", height=2, font=('times', 15, ' bold ')) lbl2.place(x=200, y=300) txt2 =
tk.Entry(window, width=20, bg="yellow", fg="red", font=('times', 25, ' bold ')) txt2.place(x=550,
y=310) clearButton = tk.Button(window, text="Clear",command=clear,fg="black" ,bg="deep
pink" ,width=10 ,height=1 ,activebackground = "Red" ,font=('times', 15, ' bold '))
clearButton.place(x=950, y=210) clearButton1 = tk.Button(window,
text="Clear",command=clear1,fg="black" ,bg="deep pink" ,width=10 ,height=1,
activebackground = "Red" ,font=('times', 15, ' bold '))
 clearButton1.place(x=950, y=310)
```

```
AP = tk.Button(window, text="Check Register
students",command=admin_panel,fg="black" ,bg="cyan" ,width=19
,height=1, activebackground = "Red" ,font=('times', 15, ' bold '))
AP.place(x=990, y=410) takeImg = tk.Button(window, text="Take
Images",command=take_img,fg="white" ,bg="blue2" ,width=20
,height=3, activebackground = "Red" ,font=('times', 15, ' bold '))
takeImg.place(x=90, y=500) trainImg = tk.Button(window, text="Train
Images",fg="black",command=trainimg ,bg="lawn green" ,width=20 ,height=3,
activebackground = "Red" ,font=('times', 15, ' bold ')) trainImg.place(x=390,
y=500) FA = tk.Button(window, text="Automatic
Attendace",fg="white",command=subjectchoose ,bg="blue2" ,width=20
,height=3, activebackground = "Red" ,font=('times', 15, ' bold ')) FA.place(x=690,
y=500) quitWindow = tk.Button(window, text="Manually Fill Attendance",
command=manually_fill ,fg="black" ,bg="lawn green" ,width=20 ,height=3,
activebackground = "Red" ,font=('times', 15, ' bold ')) quitWindow.place(x=990,
y=500) window.mainloop()
```

**testing.py**

```
import cv2
import numpy as np
recognizer = cv2.createLBPHFaceRecognizer()
recognizer.read('TrainingImageLabel/trainner.yml')
cascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath)
font = cv2.FONT_HERSHEY_SIMPLEX
cam = cv2.VideoCapture(0)
while True:
    ret, im = cam.read()
    gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(gray, 1.2, 5)
    for(x, y, w, h) in faces:
        Id, conf = recognizer.predict(gray[y:y+h, x:x+w])

        # # else:
        # #     Id="Unknown"
        # cv2.rectangle(im, (x-22,y-90), (x+w+22, y-22), (0,255,0), -1)
        cv2.rectangle(im, (x, y), (x + w, y + h), (0, 260, 0), 7)
        cv2.putText(im, str(Id), (x, y-40), font, 2, (255, 255, 255), 3)
```



```
 # cv2.putText(im, str(Id), (x + h, y), font,
1, (0, 260, 0), 2)
    cv2.imshow('im', im)
    if cv2.waitKey(10) & 0xFF == ord('q'):
        break
cam.release()
cv2.destroyAllWindows()
```

# Taining.py

```
import cv2
import os
import numpy as np
from PIL import Image
#
# recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer = cv2.face.LBPHFaceRecognizer_create()
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")


def getImagesAndLabels(path):
    # get the path of all the files in the folder
    imagePaths = [os.path.join(path, f) for f in os.listdir(path)]
    # create empth face list
    faceSamples = []
    # create empty ID list
    Ids = []
    # now looping through all the image paths and loading the Ids and the
images
```

```
for imagePath in imagePaths:
        # loading the image and converting it to gray scale
        pilImage = Image.open(imagePath).convert('L')
        # Now we are converting the PIL image into numpy array
        imageNp = np.array(pilImage, 'uint8')
        # getting the Id from the image

        Id = int(os.path.split(imagePath)[-1].split(".")[1])
        # extract the face from the training image sample
        faces = detector.detectMultiScale(imageNp)
        # If a face is there then append that in the list as well as Id of it
        for (x, y, w, h) in faces:
            faceSamples.append(imageNp[y:y+h, x:x+w])
            Ids.append(Id)
    return faceSamples, Ids


faces, Ids = getImagesAndLabels('TrainingImage')
recognizer.train(faces, np.array(Ids))
recognizer.save('TrainingImageLabel/trainner.yml')
```

# Frontend(WEBPAGES):

## •Index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
    <link rel="stylesheet" href="/styles/index.css" />
  </head>
  <body>
    <nav
      style="position: sticky; top: 0; z-index: 999; background-color: black"
    >
      <div class="navbar">
        <div>
          <a href="/index.html">FaceRecG</a>
        </div>
        <div class="nav">
          <ul id="MenuItems">
            <li><a href="/index.html">Home</a></li>
            <li><a href="#">About Us</a></li>
            <li><a href="#">Services</a></li>
            <li><a href="#">Contact</a></li>
            <li>
```

```
 <button
              class="loginbtn"
              onclick="document.getElementById('login-
form').style.display='block';
document.getElementById('contact').style.display='none';"
              style="width: auto" >
Login
            </button> </li>
          </ul>
        </div>
      </div>
    </nav>
<section
      id="contact"
      style="
        width: 100%;
        display: flex;
        align-items: center;
        justify-content: center; ">
      <div id="contact-form">
        <div>
```

```
  <h1>Nice to Meet You!</h1>
        <h4>Have a question or just want to get in touch? Let's chat.</h4>
      </div>
      <p id="failure">Oopsie...message not sent.</p>
      <p id="success">Your message was sent successfully. Thank you!</p>
<form method="post" action="/">
        <div>
          <label for="name">
            <span class="required">Name: *</span>
            <input
              type="text"
              id="name"
              name="name"
              value=""
              placeholder="Your Name"
              required="required"
              tabindex="1"
              autofocus="autofocus"
            />
          </label>
        </div>
        <div>
<label for="email">
```

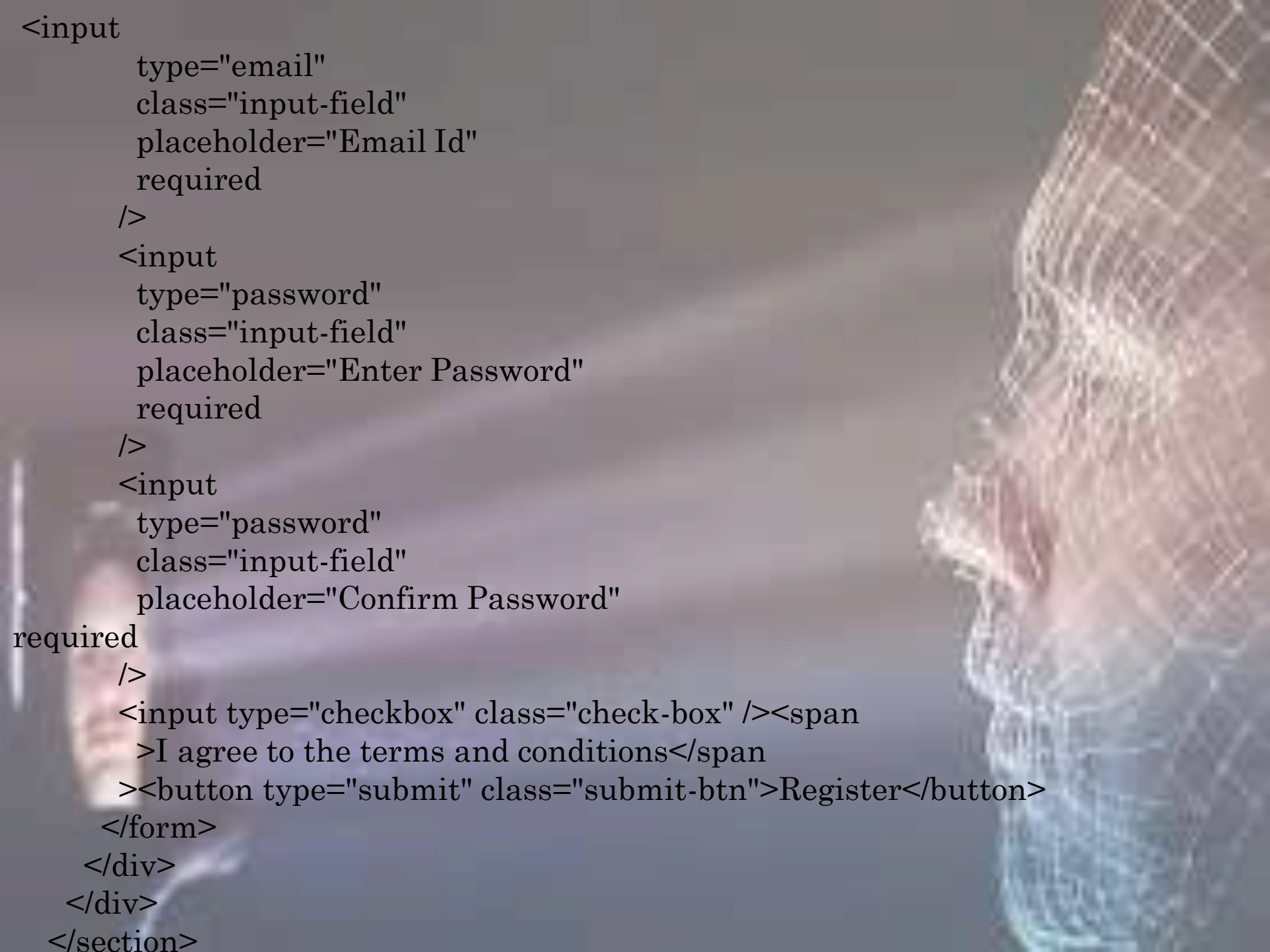
```
<span class="required">Email: *</span>
            <input
              type="email"
              id="email"
              name="email"
              value=""
              placeholder="Your Email"
tabindex="2"
              required="required"
            />
          </label>
        </div>
        <div>
          <label for="subject">
            <span>Subject: </span>
            <select id="subject" name="subject" tabindex="4">
              <option value="hello">question 1</option>
              <option value="quote">question 2</option>
              <option value="general">question 3</option>
              <option value="general">question 4</option>
            </select>
          </label>
        </div>
        <div>
```

```
<label for="message">
            <span class="required">Message: *</span>
            <textarea
id="message"
              name="message"
              placeholder="Please write your message here."
              tabindex="5"
              required="required"
            ></textarea>
          </label>
        </div>
        <div>
          <button name="submit" type="submit" id="submit">SEND</button>
        </div>
      </form>
    </div>
  </section>

  <section
    style="
      width: 100%;
      height: 80%;
      display: flex;
```

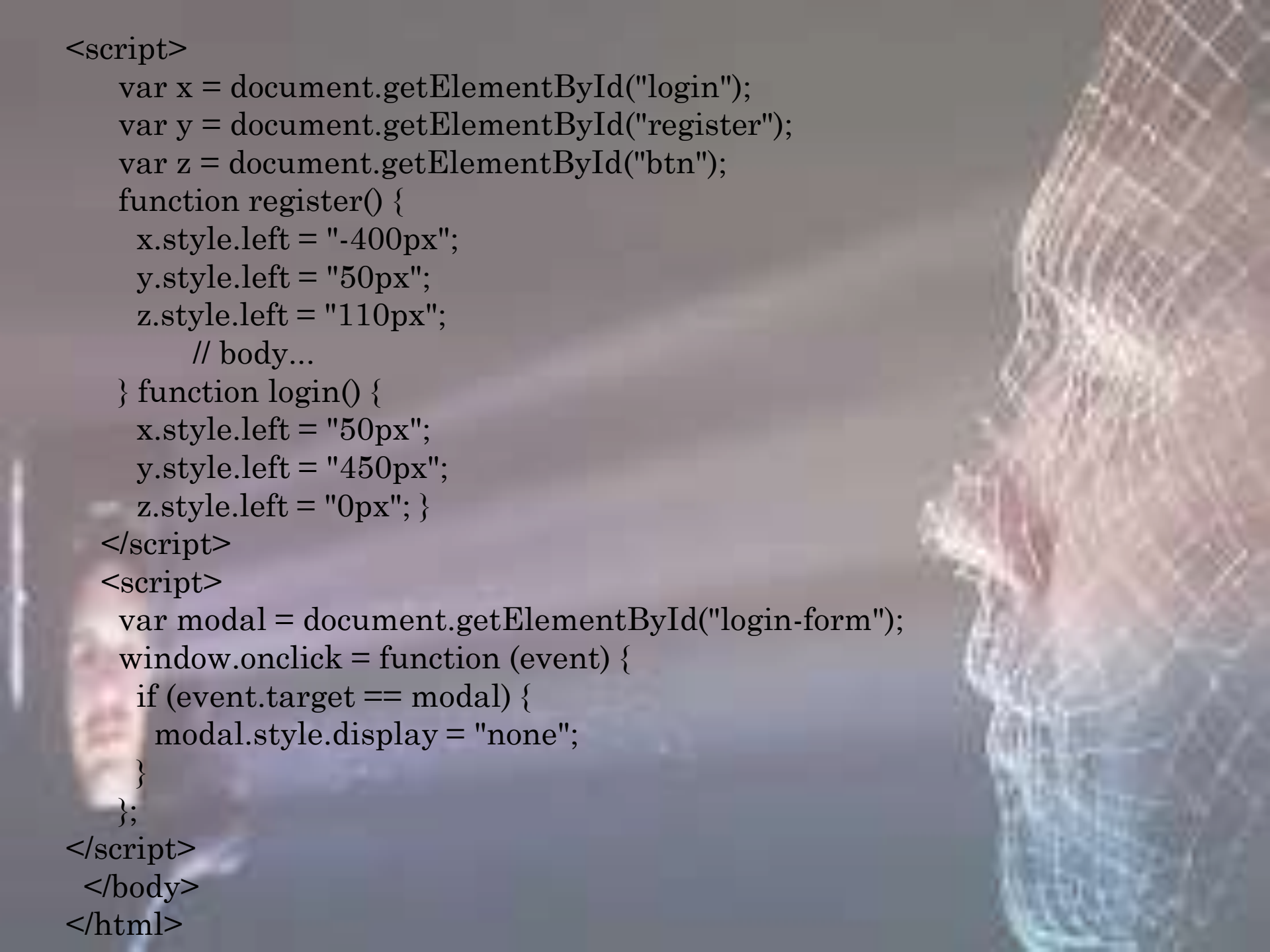
```
align-items: center;
        justify-content: center;
      "
    >
<div id="login-form" class="login-page">
        <div class="form-box">
          <div class="button-box">
            <div id="btn"></div>
            <button type="button" onclick="login()" class="toggle-btn">
              Log In
            </button>
            <button type="button" onclick="register()" class="toggle-btn">
              Register
            </button>
          </div>
          <form id="login" class="input-group-login">
            <input
              type="text"
              class="input-field"
              placeholder="Email Id"
              required
            />
```

```
<input
            type="password"
            class="input-field"
            placeholder="Enter Password"
            required
          />
          <input type="checkbox" class="check-box" /><span
            >Remember Password</span
          ><button type="submit" class="submit-btn">Log in</button>
        </form>
<form id="register" class="input-group-register">
          <input
            type="text"
            class="input-field"
            placeholder="First Name"
            required
          />
          <input
            type="text"
            class="input-field"
            placeholder="Last Name"
            required
          />
```

```
<input
          type="email"
          class="input-field"
          placeholder="Email Id"
          required
        />
        <input
          type="password"
          class="input-field"
          placeholder="Enter Password"
          required
        />
        <input
          type="password"
          class="input-field"
          placeholder="Confirm Password"
required
        />
        <input type="checkbox" class="check-box" /><span
          >I agree to the terms and conditions</span
        ><button type="submit" class="submit-btn">Register</button>
      </form>
    </div>
  </div>
</section>
```



```
<script>
    var x = document.getElementById("login");
    var y = document.getElementById("register");
    var z = document.getElementById("btn");
    function register() {
      x.style.left = "-400px";
      y.style.left = "50px";
      z.style.left = "110px";
            // body...
    } function login() {
      x.style.left = "50px";
      y.style.left = "450px";
      z.style.left = "0px"; }
  </script>
  <script>
    var modal = document.getElementById("login-form");
    window.onclick = function (event) {
      if (event.target == modal) {
        modal.style.display = "none";
      }
    };
</script>
  </body>
</html>
```



## CONTACT.php

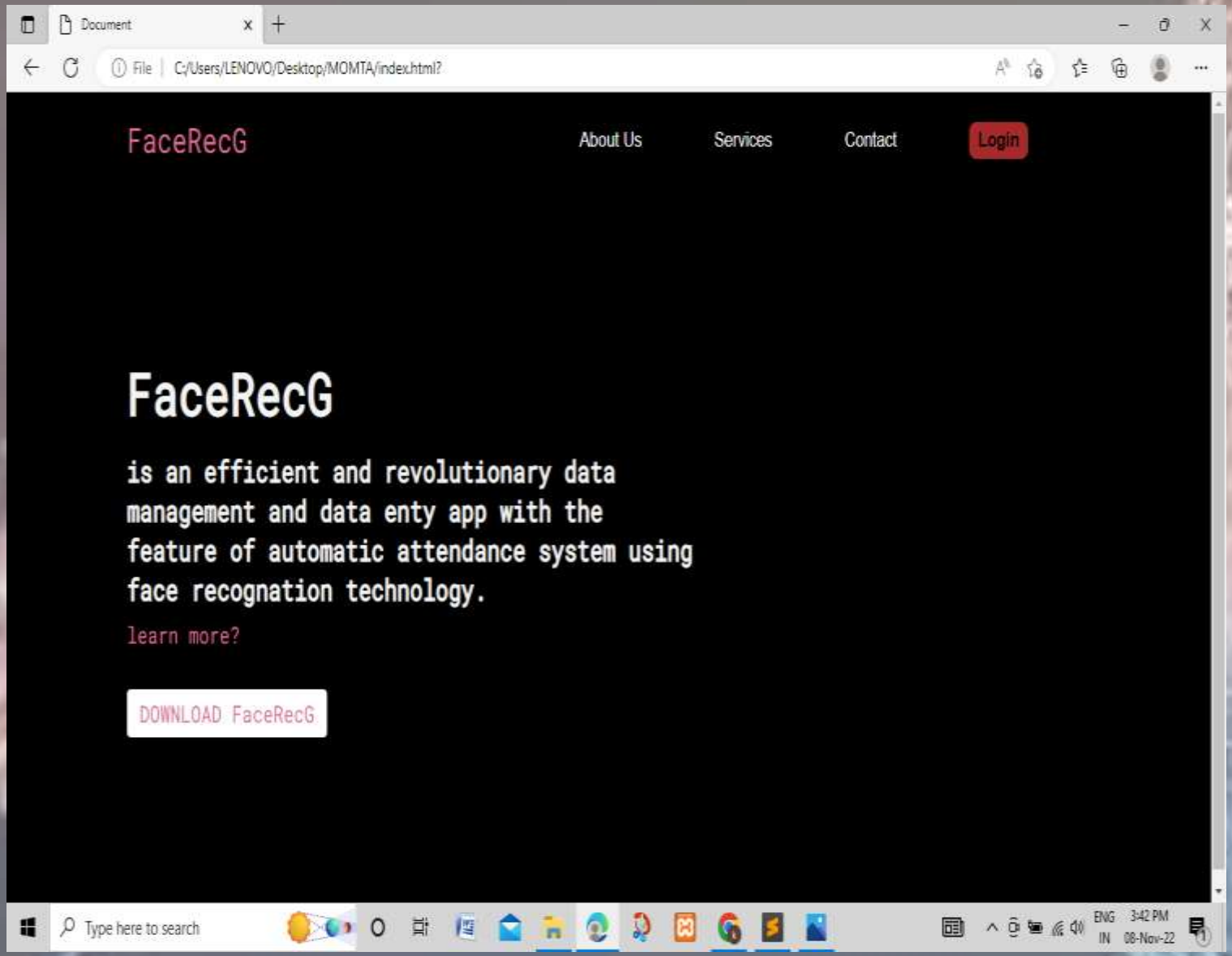
```
<?php
$firstname = filter_input(INPUT_POST, 'firstname');
$lastname = filter_input(INPUT_POST, 'lastname');
$Email = filter_input(INPUT_POST, 'Email');
$password = filter_input(INPUT_POST, 'password');
$cpassword = filter_input(INPUT_POST, 'cpassword');
if (!empty($Email)){
if (!empty($password)){
$host = "localhost";
$dbusername = "root";
$dbpassword = "";
$dbname = "login";
// Create connection
$conn = new mysqli ($host, $dbusername, $dbpassword, $dbname);
if (mysqli_connect_error()){
die('Connect Error (' . mysqli_connect_errno() .') '
. mysqli_connect_error());
}
else{
$sql = "INSERT INTO test (fn,ln,email,pass, cpass) values
('$firstname','$lastname','$Email','$password','$cpassword')";
if ($conn->query($sql)){
echo "New record is inserted sucessfully";
}
else{
echo "Error: ". $sql ."
". $conn->error;
```

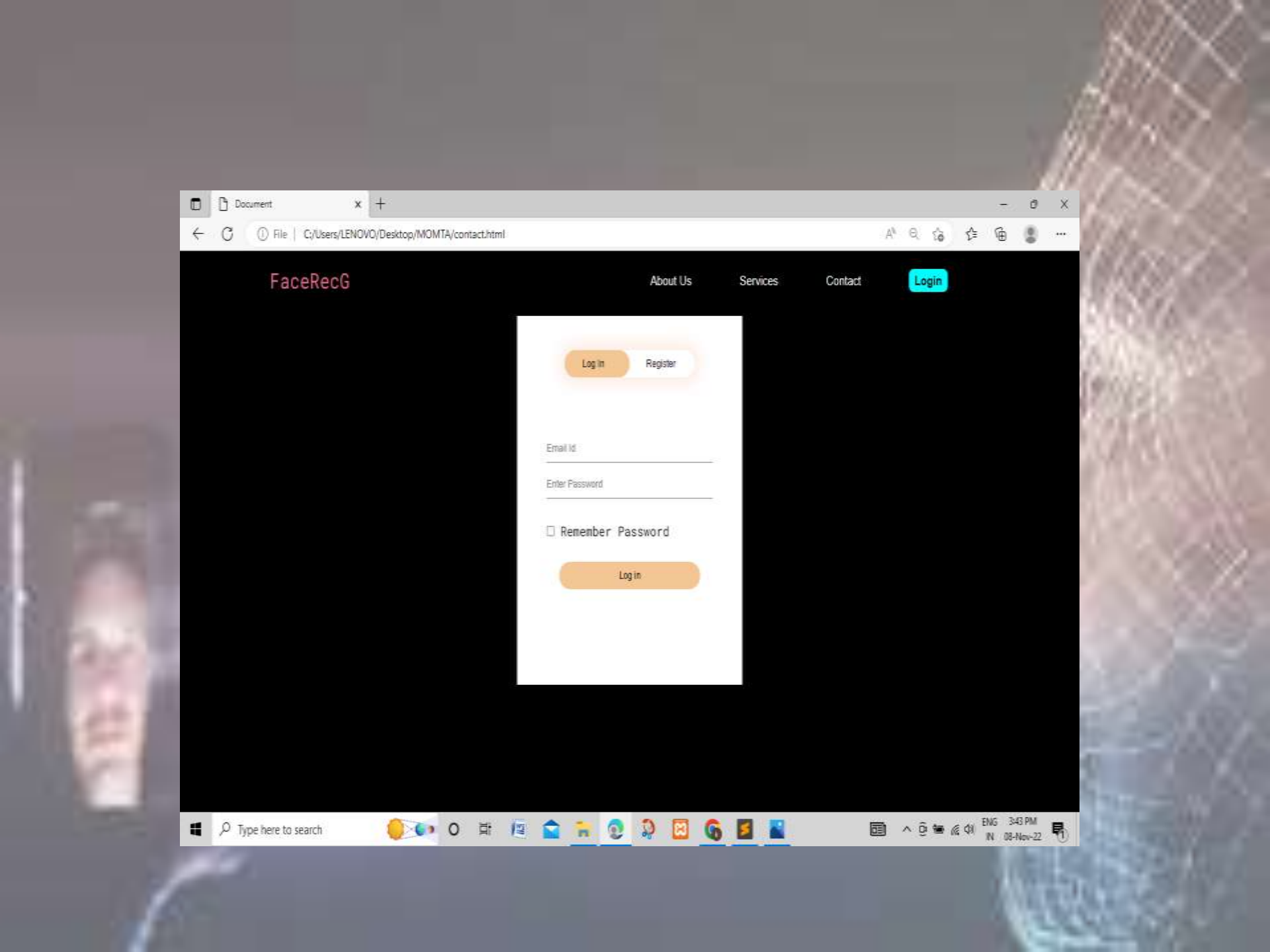
```
}
$conn->close();
}
}
else{
echo "Password should not be empty";
die();
}
}
else{
echo "Username should not be empty";
die();
}
?>
```

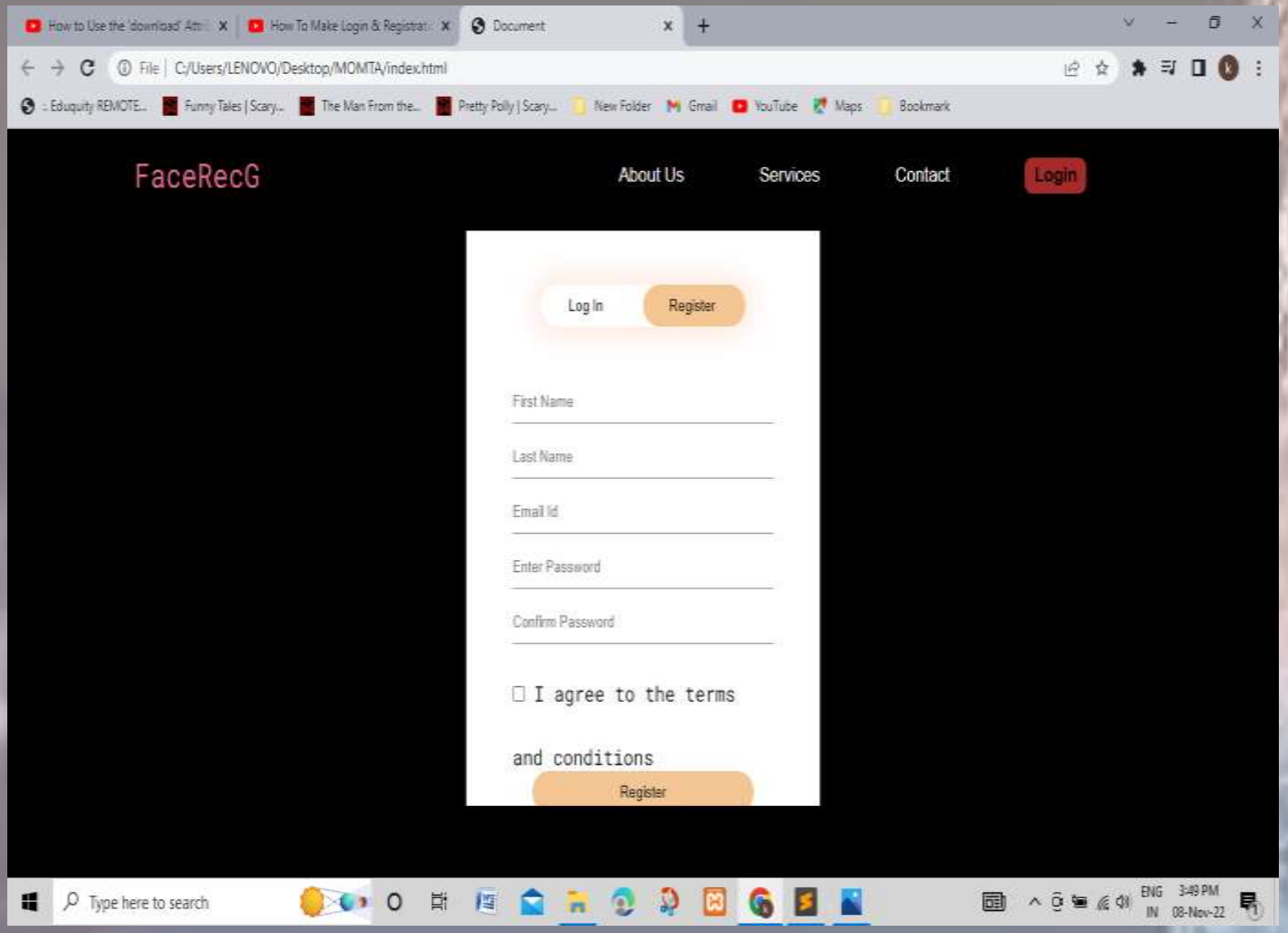
**Login.php**

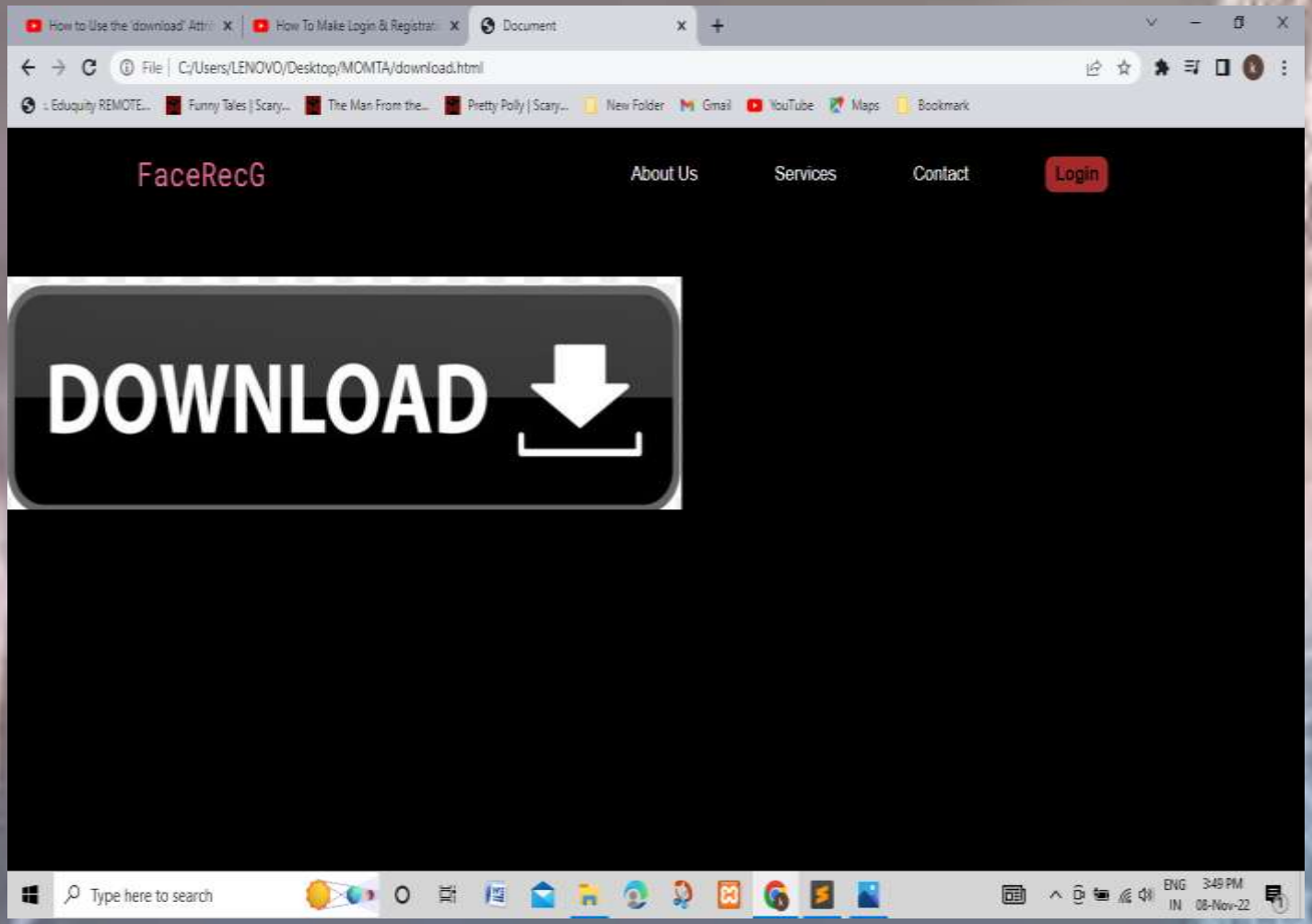
```
<?php
   $host = "localhost";
   $user = "root";
   $password = '';
   $db_name = "login";

   $con = mysqli_connect($host, $user, $password, $db_name);
   if(mysqli_connect_errno()) {
      die("Failed to connect with MySQL: ". mysqli_connect_error());
   }
?>
Log.php
<?php
   include('login.php');
   $Email = $_POST['Email'];
   $password = $_POST['password'];

      //to prevent from mysqli injection
      $Email = stripcslashes($Email);
      $password = stripcslashes($password);
      $Email = mysqli_real_escape_string($con, $Email);
      $password = mysqli_real_escape_string($con, $password);

      $sql = "SELECT * FROM test WHERE email = '$Email' AND pass = '$password'";
      $result = mysqli_query($con, $sql);
      $row = mysqli_fetch_array($result, MYSQLI_ASSOC);
      $count = mysqli_num_rows($result);

      if($count == 1){
         header("location: download.html");
      }
      else{
         echo "Login failed. Invalid username or password.";
      }
?>
```

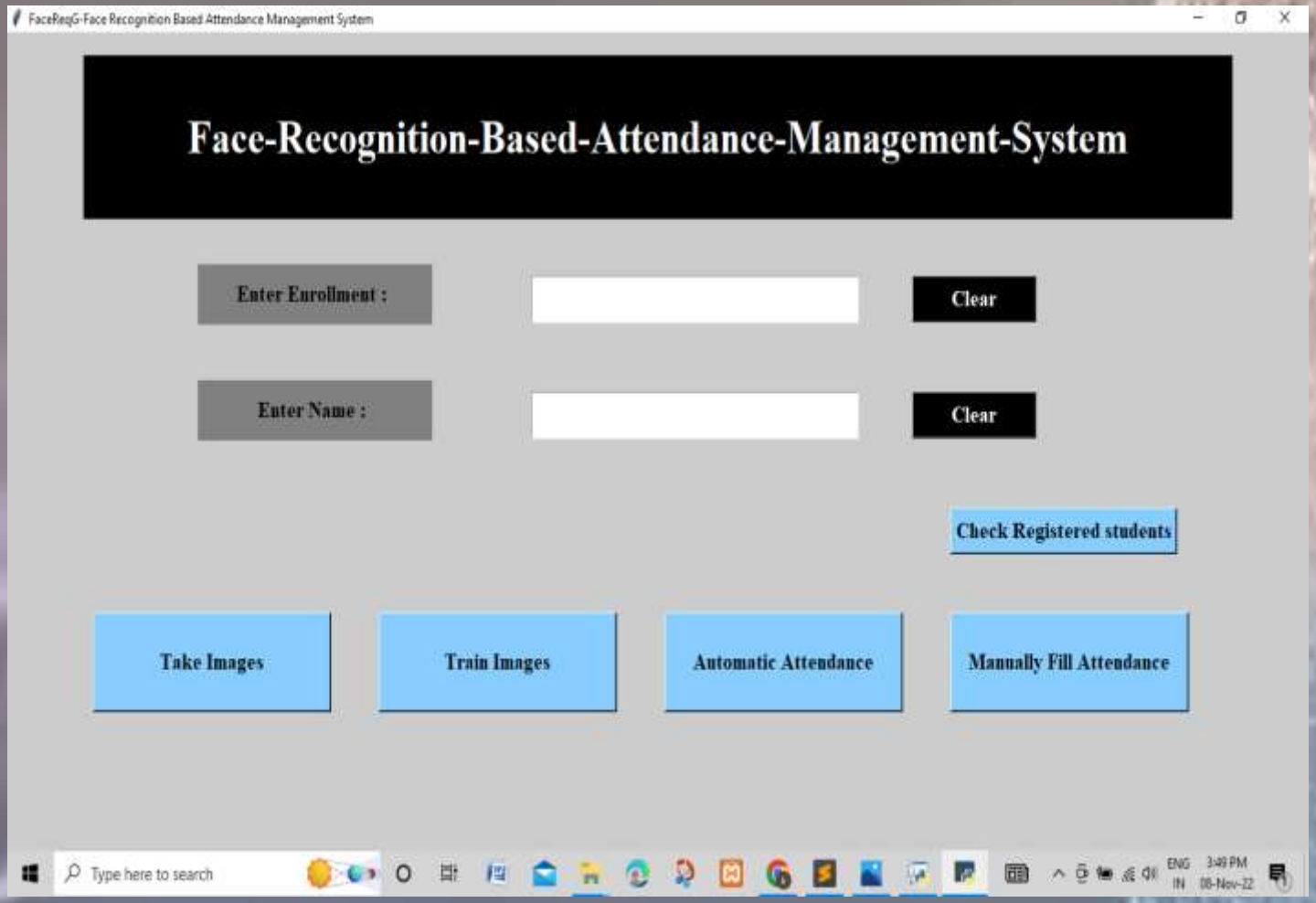
# OUTPUT

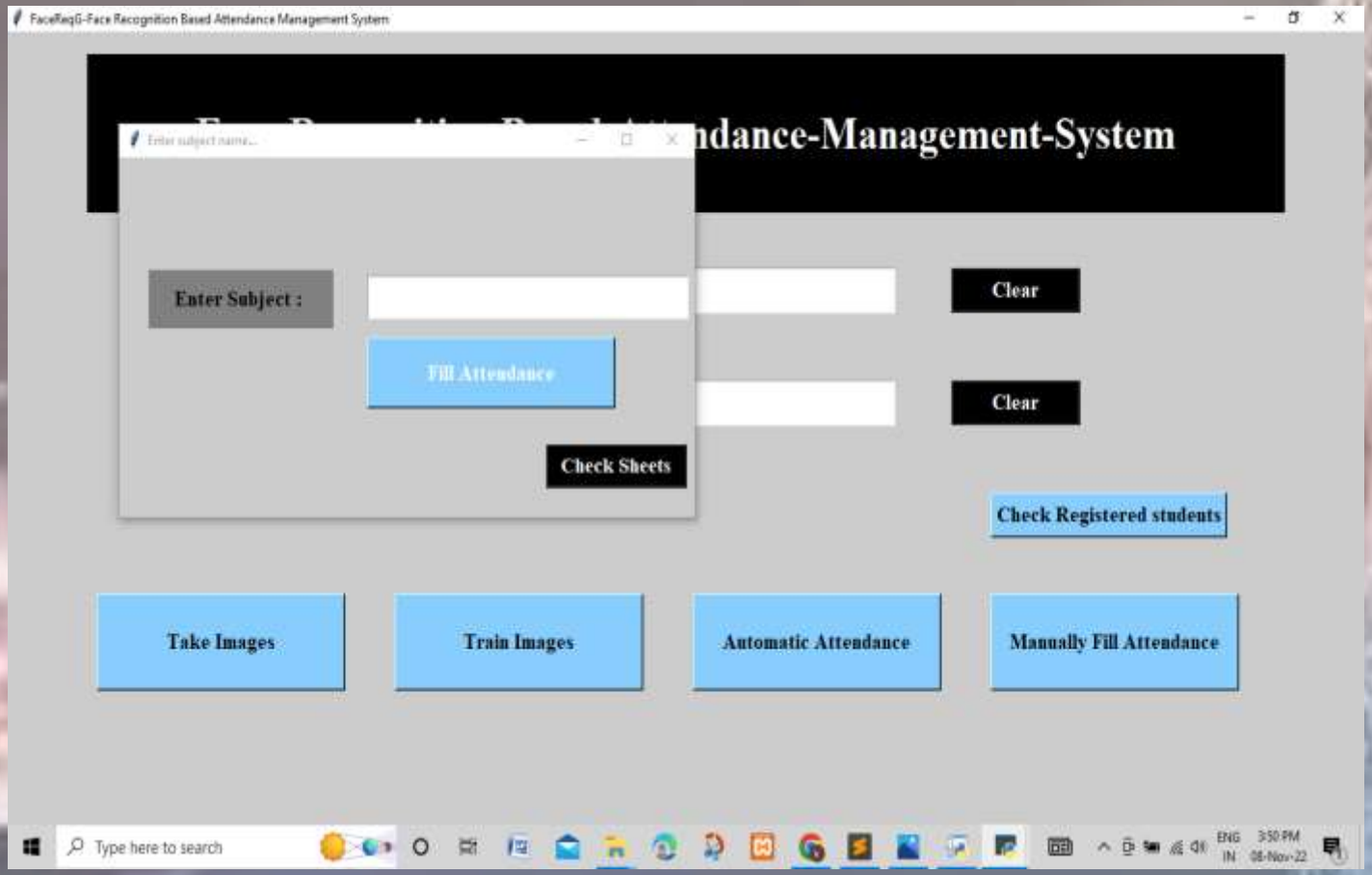

Document
File | C:/Users/LENOVO/Desktop/MOMTA/index.html?
FaceRecG
About Us
Services
Contact
Login
FaceRecG
is an efficient and revolutionary data management and data enty app with the feature of automatic attendance system using face recognation technology.
learn more?
DOWNLOAD FaceRecG
Type here to search
ENG IN
3:42 PM 08-Nov-22

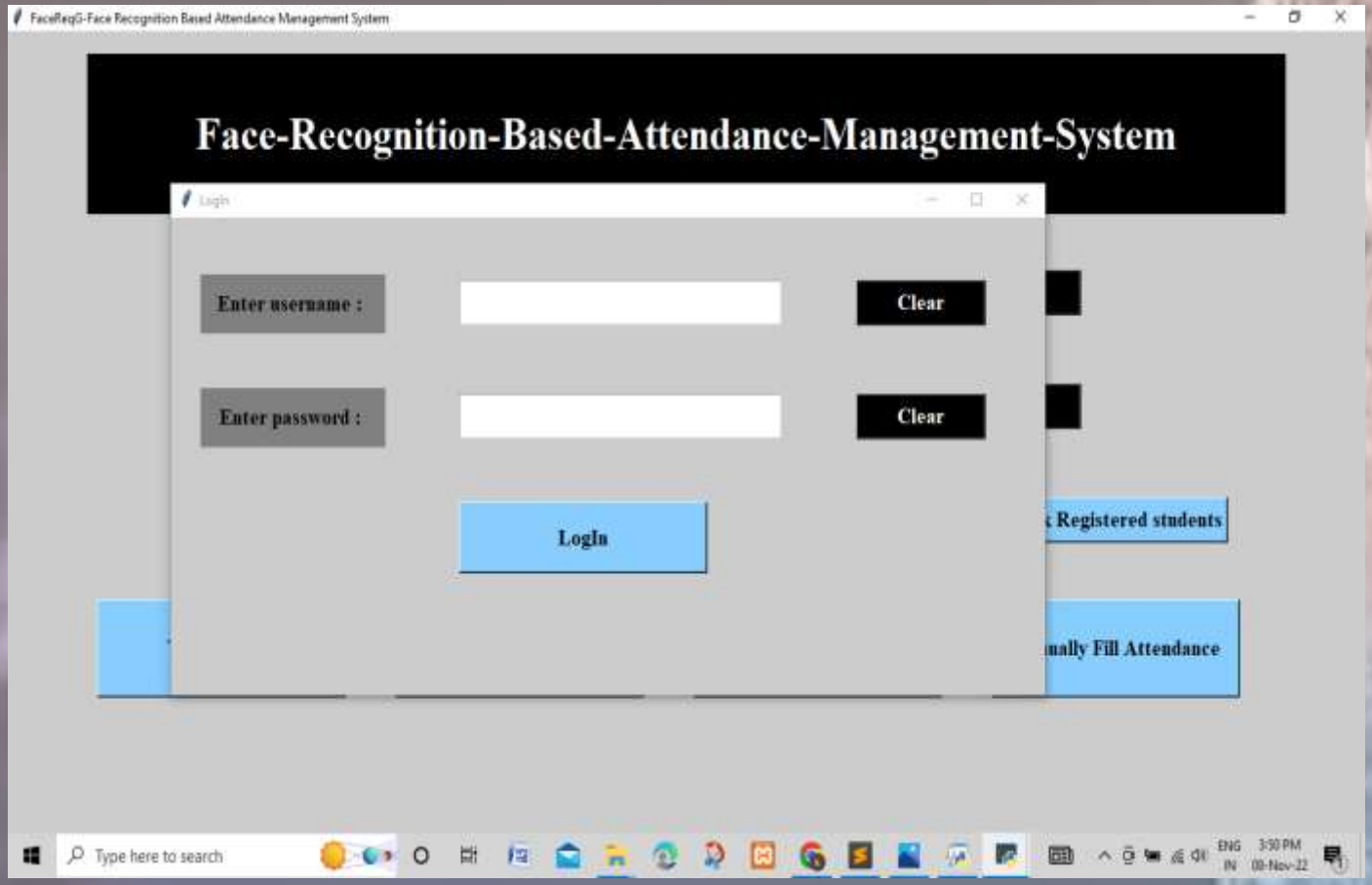

Document
File | C:/Users/LENOVO/Desktop/MOMTA/contact.html
FaceRecG
About Us
Services
Contact
Login
Log in
Register
Email Id
Enter Password
Remember Password
Log in
Type here to search
ENG
IN
3:43 PM
08-Nov-22

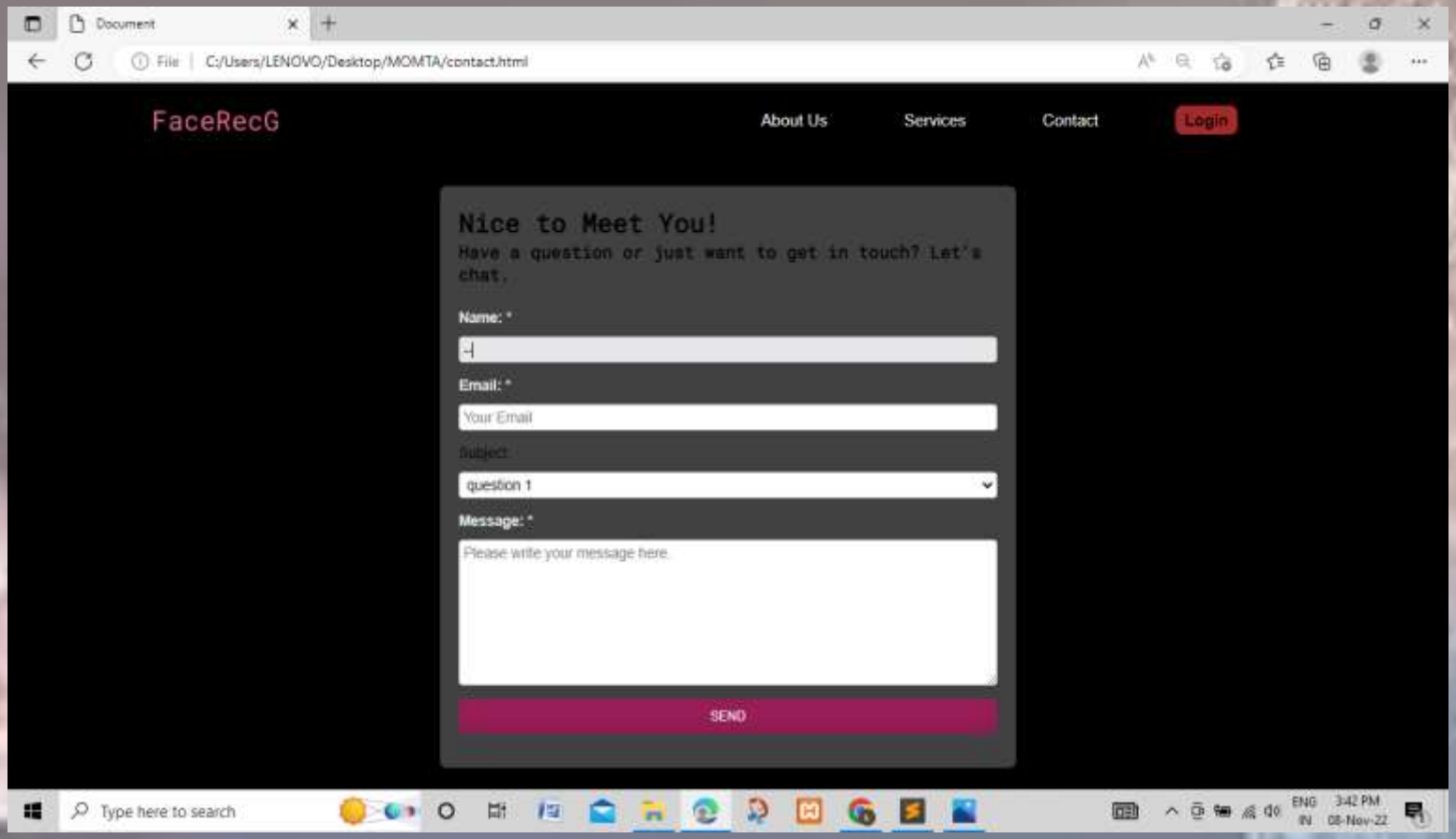

How to Use the 'download' Attr
How To Make Login & Registrati
Document
File | C:/Users/LENOVO/Desktop/MOMTA/index.html
Eduquity REMOTE...
Funny Tales | Scary...
The Man From the...
Pretty Polly | Scary...
New Folder
Gmail
YouTube
Maps
Bookmark
FaceRecG
About Us
Services
Contact
Login
Log In
Register
First Name
Last Name
Email Id
Enter Password
Confirm Password
I agree to the terms
and conditions
Register
Type here to search
ENG
IN
3:49 PM
08-Nov-22


How to Use the 'download' Attri
How To Make Login & Registrati
Document
File | C:/Users/LENOVO/Desktop/MOMTA/download.html
Eduquity REMOTE...
Funny Tales | Scary...
The Man From the...
Pretty Polly | Scary...
New Folder
Gmail
YouTube
Maps
Bookmark
FaceRecG
About Us
Services
Contact
Login
DOWNLOAD
Type here to search
ENG
IN
3:49 PM
08-Nov-22


FaceReqG-Face Recognition Based Attendance Management System
Face-Recognition-Based-Attendance-Management-System
Enter Enrollment :
Clear
Enter Name :
Clear
Check Registered students
Take Images
Train Images
Automatic Attendance
Manually Fill Attendance
Type here to search
ENG
IN
3:49 PM
08-Nov-22


FaceRecG-Face Recognition Based Attendance Management System
ndance-Management-System
Enter Subject :
Fill Attendance
Check Sheets
Clear
Clear
Check Registered students
Take Images
Train Images
Automatic Attendance
Manually Fill Attendance
Type here to search
ENG
IN
3:50 PM
08-Nov-22


FaceRegG-Face Recognition Based Attendance Management System
Face-Recognition-Based-Attendance-Management-System
LogIn
Enter username :
Clear
Enter password :
Clear
LogIn
Registered students
ually Fill Attendance
Type here to search
ENG
IN
3:50 PM
08-Nov-22


Document
File | C:/Users/LENOVO/Desktop/MOMTA/contact.html
FaceRecG
About Us
Services
Contact
Login
Nice to Meet You!
Have a question or just want to get in touch? Let's chat.
Name: *
Email: *
Your Email
Subject
question 1
Message: *
Please write your message here
SEND
Type here to search
ENG
IN
3:42 PM
08-Nov-22

# Conclusion

Thus the aim of our project is to capture the images of the students , convert it into frames , relate it with the database to ensure their presence or absence , mark attendence to the particular student to maintain the record . The automated face recognition attendance system helps in increasing the accuracy and speed ultimately achieve the high precision realtime attendance to meet the need for automatic classroom evaluation . This system is designed to minimize the human efforts for taking the attendance manually that take place in every college. As the attendance marking process is done without any human intereference , which is the main scope in the system


Thank You!

Index.html

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Document</title>
  <link rel="stylesheet" href="styles/index.css" />
 </head>

 <body>
  <nav
   style="position: sticky; top: 0; z-index: 999; background-color: black"
  >
   <div class="navbar">
    <div>
     <a href="index.html">FaceReqG</a>
    </div>
    <div class="nav">
     <ul id="MenuItems">
      <li><a href="#">About Us</a></li>
      <li><a href="#">Services</a></li>
      <li><a href="contact.html">Contact</a></li>
      <li>
       <button
```

```
            class="loginbtn"
            onclick="document.getElementById('login-form').style.display='block';
            document.getElementById('hero').style.display='none';
            document.getElementById('download').style.display='none';
            "
            style="width: auto"
           >
           Login
           </button>
          </li>
         </ul>
        </div>
       </div>
     </nav>
     <section id="hero">
      <div class="container">
       <div class="info">
        <h1>FaceReqG</h1>
        <h2>
         <ul>
          <li>efficient and evolutionary</li>
          <li>easy data entry and data management</li>
          <li>automatic attedance using face recognition technology</li>
</ul>
        </h2>
```

```
    <p>learn more?</p>
    <a href="signuplogin.html">DOWNLOAD FaceReqG</a>
    <p
     style="color: #777; font-size: 15px; bottom: 68px; margin-top: 20px"
    >
   </div>
  </div>
</section>
<section
 style="
   width: 100%;
   height: 80%;
   display: flex;
   align-items: center;
   justify-content: center;
 "
>
 <div id="login-form" class="login-page">
   <div class="form-box">
    <div class="button-box">
     <div id="btn"></div>
     <button type="button" onclick="login()" class="toggle-btn">
      Log In
     </button>
     <button type="button" onclick="register()" class="toggle-btn">
```

```
    Register
  </button>
</div>
<form id="login" class="input-group-login" action="log.php" method="post">
  <input
    type="text"
    class="input-field"
    placeholder="Email Id"
    required
    name="Email"
  />
  <input
    type="password"
    class="input-field"
    placeholder="Enter Password"
    required
    name="password"
  />
  <input type="checkbox" class="check-box" /><span
    >Remember Password</span
  ><button type="submit" class="submit-btn">Log in</a></button>
</form>
<form id="register" class="input-group-register" action="contact.php" method="post">
  <input
    type="text"
```

```
  class="input-field"
  placeholder="First Name"
  required
  name="firstname"
/>
<input
  type="text"
  class="input-field"
  placeholder="Last Name"
  required
  name="lastname"
/>
<input
  type="email"
  class="input-field"
  placeholder="Email Id"
  required
  name="Email"
/>
<input
  type="password"
  class="input-field"
  placeholder="Enter Password"
  required
  name="password"
```

```
      />
      <input
        type="password"
        class="input-field"
        placeholder="Confirm Password"
        required
        name="cpassword"
      />
      <input type="checkbox" class="check-box" /><span
        >I agree to the terms and conditions</span
      ><button type="submit" class="submit-btn" name="submit">Register</button>
    </form>
  </div>
 </div>
</section>

<script>
 var x = document.getElementById("login");
 var y = document.getElementById("register");
 var z = document.getElementById("btn");
 var d = document.getElementById("download");

 function register() {
  x.style.left = "-400px";
  y.style.left = "50px";
```

```
      z.style.left = "110px";


      // body...
    }
    function login() {
      x.style.left = "50px";
      y.style.left = "450px";
      z.style.left = "0px";
    }
  </script>
  <script>
    var modal = document.getElementById("login-form");
    window.onclick = function (event) {
      if (event.target == modal) {
        modal.style.display = "none";
      }
    };
  </script>
 </body>
</html>
```

CONTACT.HTML

```
<?php


$firstname = filter_input(INPUT_POST, 'firstname');
```

```
$lastname = filter_input(INPUT_POST, 'lastname');

$Email = filter_input(INPUT_POST, 'Email');

$password = filter_input(INPUT_POST, 'password');

$cpassword = filter_input(INPUT_POST, 'cpassword');



if (!empty($Email)){

if (!empty($password)){

$host = "localhost";

$dbusername = "root";

$dbpassword = "";

$dbname = "login";

// Create connection

$conn = new mysqli ($host, $dbusername, $dbpassword, $dbname);



if (mysqli_connect_error()){

die('Connect Error (. mysqli_connect_errno() .') '

. mysqli_connect_error());

}

else{

$sql = "INSERT INTO test (fn,ln,email,pass, cpass) values
('$firstname','$lastname','$Email','$password','$cpassword')";

if ($conn->query($sql)){

echo "New record is inserted sucessfully";

}
```

```
else{
echo "Error: ". $sql ."
". $conn->error;
}
$conn->close();
}
}
else{
echo "Password should not be empty";
die();
}
}
else{
echo "Username should not be empty";
die();
}
?>
```

Contact.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
```

```
 <link rel="stylesheet" href="styles/index.css" />
</head>
<body>
 <nav
  style="position: sticky; top: 0; z-index: 999; background-color: black"
 >
  <div class="navbar">
   <div>
    <a href="index.html">FaceRecG</a>
   </div>
   <div class="nav">
    <ul id="MenuItems">
     <li><a href="#">About Us</a></li>
     <li><a href="#">Services</a></li>
     <li><a href="contact.html">Contact</a></li>
     <li>
      <button
       class="loginbtn"
       onclick="document.getElementById('login-form').style.display='block';
       document.getElementById('contact').style.display='none';
       document.getElementById('hero').style.display='none';"
       style="width: auto"
      >
       Login
      </button>
```

```
      </li>
     </ul>
    </div>
   </div>
</nav>

<section
  id="contact"
  style="
    width: 100%;
    display: flex;
    align-items: center;
    justify-content: center;
  "
>
  <div id="contact-form">
    <div>
      <h1>Nice to Meet You!</h1>
      <h4>Have a question or just want to get in touch? Let's chat.</h4>
    </div>
    <p id="failure">Oopsie...message not sent.</p>
    <p id="success">Your message was sent successfully. Thank you!</p>

    <form method="post" action="/">
      <div>
```

```
  <label for="name">
    <span class="required">Name: *</span>
    <input
      type="text"
      id="name"
      name="name"
      value=""
      placeholder="Your Name"
      required="required"
      tabindex="1"
      autofocus="autofocus"
    />
  </label>
</div>
<div>
  <label for="email">
    <span class="required">Email: *</span>
    <input
      type="email"
      id="email"
      name="email"
      value=""
      placeholder="Your Email"
      tabindex="2"
      required="required"
```

```
    />
  </label>
</div>
<div>
  <label for="subject">
    <span>Subject: </span>
    <select id="subject" name="subject" tabindex="4">
      <option value="hello">question 1</option>
      <option value="quote">question 2</option>
      <option value="general">question 3</option>
      <option value="general">question 4</option>
    </select>
  </label>
</div>
<div>
  <label for="message">
    <span class="required">Message: *</span>
    <textarea
      id="message"
      name="message"
      placeholder="Please write your message here."
      tabindex="5"
      required="required"
    ></textarea>
  </label>
```

```
    </div>
    <div>
     <button name="submit" type="submit" id="submit">SEND</button>
    </div>
   </form>
  </div>
</section>

<section
 style="
   width: 100%;
   height: 80%;
   display: flex;
   align-items: center;
   justify-content: center;
 "
>
 <div id="login-form" class="login-page">
   <div class="form-box">
    <div class="button-box">
     <div id="btn"></div>
     <button type="button" onclick="login()" class="toggle-btn">
       Log In
     </button>
     <button type="button" onclick="register()" class="toggle-btn">
```

```
    Register
  </button>
</div>
<form id="login" class="input-group-login">
  <input
    type="text"
    class="input-field"
    placeholder="Email Id"
    required
  />
  <input
    type="password"
    class="input-field"
    placeholder="Enter Password"
    required
  />
  <input type="checkbox" class="check-box" /><span
    >Remember Password</span
  ><button type="submit" class="submit-btn">Log in</button>
</form>
<form id="register" class="input-group-register">
  <input
    type="text"
    class="input-field"
    placeholder="First Name"
```

```
  required
/>
<input
  type="text"
  class="input-field"
  placeholder="Last Name"
  required
/>
<input
  type="email"
  class="input-field"
  placeholder="Email Id"
  required
/>
<input
  type="password"
  class="input-field"
  placeholder="Enter Password"
  required
/>
<input
  type="password"
  class="input-field"
  placeholder="Confirm Password"
  required
```

```
      />
      <input type="checkbox" class="check-box" /><span
        >I agree to the terms and conditions</span
      ><button type="submit" class="submit-btn">Register</button>
    </form>
  </div>
 </div>
</section>

<script>
 var x = document.getElementById("login");
 var y = document.getElementById("register");
 var z = document.getElementById("btn");
 function register() {
  x.style.left = "-400px";
  y.style.left = "50px";
  z.style.left = "110px";

  // body...
 }
 function login() {
  x.style.left = "50px";
  y.style.left = "450px";
  z.style.left = "0px";
 }
```

```
  </script>
  <script>
    var modal = document.getElementById("login-form");
    window.onclick = function (event) {
      if (event.target == modal) {
        modal.style.display = "none";
      }
    };
  </script>
 </body>
</html>
```

Login.php

```
<?php
  $host = "localhost";
  $user = "root";
  $password = '';
  $db_name = "login";

  $con = mysqli_connect($host, $user, $password, $db_name);
  if(mysqli_connect_errno()) {
     die("Failed to connect with MySQL: ". mysqli_connect_error());
  }
```

```
?>
```

Log.php

```
<?php
   include('login.php');
   $Email = $_POST['Email'];
   $password = $_POST['password'];

      //to prevent from mysqli injection
      $Email = stripcslashes($Email);
      $password = stripcslashes($password);
      $Email = mysqli_real_escape_string($con, $Email);
      $password = mysqli_real_escape_string($con, $password);

      $sql = "SELECT * FROM test WHERE email = '$Email' AND pass = '$password'";
      $result = mysqli_query($con, $sql);
      $row = mysqli_fetch_array($result, MYSQLI_ASSOC);
      $count = mysqli_num_rows($result);

      if($count == 1){
         header("location: download.html");
      }
      else{
         echo "Login failed. Invalid username or password.";
```

```
    }
?>
```

Signuplogin.html

```
<!DOCTYPE html>
<html>
<head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>download facereqg</title>
</head>
<body>
        <h1>Login/SignUp to download FaceReqG app</h1>
        <a href="index.html">Click on "LOGIN" button to SignUp or Login.</a>



</body>
</html>
```

Download.html

```
<!DOCTYPE html>
<html lang="en">
 <head>
   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

```
  <title>Document</title>
  <link rel="stylesheet" href="styles/index.css" />
</head>

<body>
  <nav
    style="position: sticky; top: 0; z-index: 999; background-color: black"
  >
    <div class="navbar">
      <div>
        <a href="index.html">FaceRecG</a>
      </div>
      <div class="nav">
        <ul id="MenuItems">
          <li><a href="#">About Us</a></li>
          <li><a href="#">Services</a></li>
          <li><a href="contact.html">Contact</a></li>
          <li>
            <button
              class="loginbtn"
              onclick="document.getElementById('login-form').style.display='block';
              document.getElementById('hero').style.display='none';
              document.getElementById('contact').style.display='none';"
              style="width: auto"
            >
```

```
            Login
          </button>
        </li>
      </ul>
    </div>
  </div>
</nav>

<body>
  <h1>FaceReqG</h1>
    <a href=".\FaceReqG.rar" download class="download-btn"><img src="download.jpg"><i class="fa fa-download"></i></a>
</body>

<script>
  var x = document.getElementById("login");
  var y = document.getElementById("register");
  var z = document.getElementById("btn");
  function register() {
    x.style.left = "-400px";
    y.style.left = "50px";
    z.style.left = "110px";

    // body...
```

```
    }
    function login() {
      x.style.left = "50px";
      y.style.left = "450px";
      z.style.left = "0px";
    }
  </script>
  <script>
    var modal = document.getElementById("login-form");
    window.onclick = function (event) {
      if (event.target == modal) {
        modal.style.display = "none";
      }
    };
  </script>
 </body>
</html>
```

Login ad register.html

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Document</title>
  <link rel="stylesheet" href="styles/index.css" />
```

```
</head>
<body>
 <nav
  style="position: sticky; top: 0; z-index: 999; background-color: black"
 >
  <div class="navbar">
   <div>
    <a href="index.html">FaceRecG</a>
   </div>
   <div class="nav">
    <ul id="MenuItems">
     <li><a href="#">About Us</a></li>
     <li><a href="#">Services</a></li>
     <li><a href="contact.html">Contact</a></li>
     <li>
      <button
       class="loginbtn"
       onclick="document.getElementById('login-form').style.display='block';
       document.getElementById('contact').style.display='none';
       document.getElementById('hero').style.display='none';
       "style="width: auto"
      >
       Login
      </button>
     </li>
```

```
      </ul>
    </div>
  </div>
</nav>

<section
  style="
    width: 100%;
    height: 80%;
    display: flex;
    align-items: center;
    justify-content: center;
  "
>
  <div id="login-form" class="login-page">
    <div class="form-box">
      <div class="button-box">
        <div id="btn"></div>
        <button type="button" onclick="login()" class="toggle-btn">
          Log In
        </button>
        <button type="button" onclick="register()" class="toggle-btn">
          Register
        </button>
      </div>
```

```
<form id="login" class="input-group-login">
 <input
  type="text"
  class="input-field"
  placeholder="Email Id"
  required
 />
 <input
  type="password"
  class="input-field"
  placeholder="Enter Password"
  required
 />
 <input type="checkbox" class="check-box" /><span
  >Remember Password</span
 ><button type="submit" class="submit-btn">Log in</button>
</form>
<form id="register" class="input-group-register">
 <input
  type="text"
  class="input-field"
  placeholder="First Name"
  required
 />
 <input
```

```
  type="text"
  class="input-field"
  placeholder="Last Name"
  required
/>
<input
  type="email"
  class="input-field"
  placeholder="Email Id"
  required
/>
<input
  type="password"
  class="input-field"
  placeholder="Enter Password"
  required
/>
<input
  type="password"
  class="input-field"
  placeholder="Confirm Password"
  required
/>
<input type="checkbox" class="check-box" /><span
  >I agree to the terms and conditions</span
```

```
      ><button type="submit" class="submit-btn">Register</button>
    </form>
   </div>
  </div>
</section>

<script>
  var x = document.getElementById("login");
  var y = document.getElementById("register");
  var z = document.getElementById("btn");
  function register() {
    x.style.left = "-400px";
    y.style.left = "50px";
    z.style.left = "110px";

    // body...
  }
  function login() {
    x.style.left = "50px";
    y.style.left = "450px";
    z.style.left = "0px";
  }
</script>
<script>
  var modal = document.getElementById("login-form");
```

```
    window.onclick = function (event) {
      if (event.target == modal) {
        modal.style.display = "none";
      }
    };
  </script>
 </body>
</html>
```

## Main.py

**Source code**:

```
import tkinter as tk
from tkinter import *
import cv2
import csv
import os
import numpy as np
from PIL import Image, ImageTk
import pandas as pd
import datetime
import time

# Window is our Main frame of system
window = tk.Tk()
window.title("FaceReqG-Face Recognition Based Attendance Management System")

window.geometry('1280x720')
window.configure(background='grey80')

# GUI for manually fill attendance
```

```
def manually_fill():
    global sb
    sb = tk.Tk()
    # sb.iconbitmap('AMS.ico')
    sb.title("Enter subject name...")
    sb.geometry('580x320')
    sb.configure(background='grey80')

    def err_screen_for_subject():

        def ec_delete():
            ec.destroy()
        global ec
        ec = tk.Tk()
        ec.geometry('300x100')
        # ec.iconbitmap('AMS.ico')
        ec.title('Warning!!')
        ec.configure(background='snow')
        Label(ec, text='Please enter your subject name!!!', fg='red',
              bg='white', font=('times', 16, ' bold ')).pack()
        Button(ec, text='OK', command=ec_delete, fg="black", bg="lawn green", width=9, height=1,
activebackground="Red",
               font=('times', 15, ' bold ')).place(x=90, y=50)

    def fill_attendance():
        ts = time.time()
```

```
Date = datetime.datetime.fromtimestamp(ts).strftime('%Y_%m_%d')
timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
Time = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
Hour, Minute, Second = timeStamp.split(":")
# Creatting csv of attendance

# Create table for Attendance
date_for_DB = datetime.datetime.fromtimestamp(ts).strftime('%Y_%m_%d')
global subb
subb = SUB_ENTRY.get()
DB_table_name = str(subb + "_" + Date + "_Time_" +
                    Hour + "_" + Minute + "_" + Second)

import pymysql.connections

# Connect to the database
try:
   global cursor
   connection = pymysql.connect(
      host='localhost', user='root', password='', db='manually_fill_attendance')
   cursor = connection.cursor()
except Exception as e:
   print(e)

sql = "CREATE TABLE " + DB_table_name + """
```

```
            (ID INT NOT NULL AUTO_INCREMENT,
             ENROLLMENT varchar(100) NOT NULL,
             NAME VARCHAR(50) NOT NULL,
             DATE VARCHAR(20) NOT NULL,
             TIME VARCHAR(20) NOT NULL,
                PRIMARY KEY (ID)
                );
            """


try:
    cursor.execute(sql)  # for create a table
except Exception as ex:
    print(ex)  #


if subb == '':
    err_screen_for_subject()
else:
    sb.destroy()
    MFW = tk.Tk()
    # MFW.iconbitmap('AMS.ico')
    MFW.title("Manually attendance of " + str(subb))
    MFW.geometry('880x470')
    MFW.configure(background='grey80')


    def del_errsc2():
```

```
    errsc2.destroy()


def err_screen1():
    global errsc2
    errsc2 = tk.Tk()
    errsc2.geometry('330x100')
    # errsc2.iconbitmap('AMS.ico')
    errsc2.title('Warning!!')
    errsc2.configure(background='grey80')
    Label(errsc2, text='Please enter Student & Enrollment!!!', fg='black', bg='white',
          font=('times', 16, ' bold ')).pack()
    Button(errsc2, text='OK', command=del_errsc2, fg="black", bg="lawn green", width=9,
height=1,
           activebackground="Red", font=('times', 15, ' bold ')).place(x=90, y=50)


def testVal(inStr, acttyp):
    if acttyp == '1':  # insert
        if not inStr.isdigit():
            return False
    return True


ENR = tk.Label(MFW, text="Enter Enrollment", width=15, height=2, fg="black", bg="grey",
               font=('times', 15))
ENR.place(x=30, y=100)
```

```
    STU_NAME = tk.Label(MFW, text="Enter Student name", width=15, height=2, fg="black",
bg="grey",
                        font=('times', 15))
    STU_NAME.place(x=30, y=200)


    global ENR_ENTRY
    ENR_ENTRY = tk.Entry(MFW, width=20, validate='key',
                         bg="white", fg="black", font=('times', 23))
    ENR_ENTRY['validatecommand'] = (
       ENR_ENTRY.register(testVal), '%P', '%d')
    ENR_ENTRY.place(x=290, y=105)


    def remove_enr():
       ENR_ENTRY.delete(first=0, last=22)


    STUDENT_ENTRY = tk.Entry(
       MFW, width=20, bg="white", fg="black", font=('times', 23))
    STUDENT_ENTRY.place(x=290, y=205)


    def remove_student():
       STUDENT_ENTRY.delete(first=0, last=22)


    # get important variable
    def enter_data_DB():
       ENROLLMENT = ENR_ENTRY.get()
       STUDENT = STUDENT_ENTRY.get()
```

```
    if ENROLLMENT == '':
        err_screen1()
    elif STUDENT == '':
        err_screen1()
    else:
        time = datetime.datetime.fromtimestamp(
            ts).strftime('%H:%M:%S')
        Hour, Minute, Second = time.split(":")
        Insert_data = "INSERT INTO " + DB_table_name + \
            " (ID,ENROLLMENT,NAME,DATE,TIME) VALUES (0, %s, %s, %s,%s)"
        VALUES = (str(ENROLLMENT), str(
            STUDENT), str(Date), str(time))
        try:
            cursor.execute(Insert_data, VALUES)
        except Exception as e:
            print(e)
        ENR_ENTRY.delete(first=0, last=22)
        STUDENT_ENTRY.delete(first=0, last=22)


def create_csv():
    import csv
    cursor.execute("select * from " + DB_table_name + ";")
    csv_name = 'Attendance/Manually Attendance/'+DB_table_name+'.csv'
    with open(csv_name, "w") as csv_file:
        csv_writer = csv.writer(csv_file)
```

```
    csv_writer.writerow(
        [i[0] for i in cursor.description])  # write headers
    csv_writer.writerows(cursor)
    O = "CSV created Successfully"
    Notifi.configure(text=O, bg="Green", fg="white",
                     width=33, font=('times', 19, 'bold'))
    Notifi.place(x=180, y=380)
import csv
import tkinter
root = tkinter.Tk()
root.title("Attendance of " + subb)
root.configure(background='grey80')
with open(csv_name, newline="") as file:
    reader = csv.reader(file)
    r = 0


    for col in reader:
        c = 0
        for row in col:
            # i've added some styling
            label = tkinter.Label(root, width=18, height=1, fg="black", font=('times', 13, ' bold '),
                                  bg="white", text=row, relief=tkinter.RIDGE)
            label.grid(row=r, column=c)
            c += 1
        r += 1
```

```
    root.mainloop()


Notifi = tk.Label(MFW, text="CSV created Successfully", bg="Green", fg="white", width=33,
                  height=2, font=('times', 19, 'bold'))


c1ear_enroll = tk.Button(MFW, text="Clear", command=remove_enr, fg="white", bg="black",
width=10,
                         height=1,
                         activebackground="white", font=('times', 15, ' bold '))
c1ear_enroll.place(x=690, y=100)


c1ear_student = tk.Button(MFW, text="Clear", command=remove_student, fg="white",
bg="black", width=10,
                          height=1,
                          activebackground="white", font=('times', 15, ' bold '))
c1ear_student.place(x=690, y=200)


DATA_SUB = tk.Button(MFW, text="Enter Data", command=enter_data_DB, fg="black",
bg="SkyBlue1", width=20,
                     height=2,
                     activebackground="white", font=('times', 15, ' bold '))
DATA_SUB.place(x=170, y=300)


MAKE_CSV = tk.Button(MFW, text="Convert to CSV", command=create_csv, fg="black",
bg="SkyBlue1", width=20,
                     height=2,
                     activebackground="white", font=('times', 15, ' bold '))
```

```
        MAKE_CSV.place(x=570, y=300)


        def attf():
            import subprocess
            subprocess.Popen(
                r'explorer /select,"Attendance\Manually Attendance\"')


        attf = tk.Button(MFW,  text="Check Sheets", command=attf, fg="white", bg="black",
                         width=12, height=1, activebackground="white", font=('times', 14, ' bold '))
        attf.place(x=730, y=410)


        MFW.mainloop()


    SUB = tk.Label(sb, text="Enter Subject : ", width=15, height=2,
                   fg="black", bg="grey80", font=('times', 15, ' bold '))
    SUB.place(x=30, y=100)


    global SUB_ENTRY


    SUB_ENTRY = tk.Entry(sb, width=20, bg="white",
                         fg="black", font=('times', 23))
    SUB_ENTRY.place(x=250, y=105)


    fill_manual_attendance = tk.Button(sb, text="Fill Attendance", command=fill_attendance, fg="black",
bg="SkyBlue1", width=20, height=2,
                                       activebackground="white", font=('times', 15, ' bold '))
```

```
    fill_manual_attendance.place(x=250, y=160)
    sb.mainloop()

# For clear textbox


def clear():
    txt.delete(first=0, last=22)


def clear1():
    txt2.delete(first=0, last=22)


def del_sc1():
    sc1.destroy()


def err_screen():
    global sc1
    sc1 = tk.Tk()
    sc1.geometry('300x100')
    # sc1.iconbitmap('AMS.ico')
    sc1.title('Warning!!')
    sc1.configure(background='grey80')
```

```
    Label(sc1, text='Enrollment & Name required!!!', fg='black',
          bg='white', font=('times', 16)).pack()
    Button(sc1, text='OK', command=del_sc1, fg="black", bg="lawn green", width=9,
           height=1, activebackground="Red", font=('times', 15, ' bold ')).place(x=90, y=50)

# Error screen2



def del_sc2():
    sc2.destroy()



def err_screen1():
    global sc2
    sc2 = tk.Tk()
    sc2.geometry('300x100')
    # sc2.iconbitmap('AMS.ico')
    sc2.title('Warning!!')
    sc2.configure(background='grey80')
    Label(sc2, text='Please enter your subject name!!!', fg='black',
          bg='white', font=('times', 16)).pack()
    Button(sc2, text='OK', command=del_sc2, fg="black", bg="lawn green", width=9,
           height=1, activebackground="Red", font=('times', 15, ' bold ')).place(x=90, y=50)

# For take images for datasets
```

```
def take_img():
    l1 = txt.get()
    l2 = txt2.get()
    if l1 == '':
        err_screen()
    elif l2 == '':
        err_screen()
    else:
        try:
            cam = cv2.VideoCapture(0)
            detector = cv2.CascadeClassifier(
                'haarcascade_frontalface_default.xml')
            Enrollment = txt.get()
            Name = txt2.get()
            sampleNum = 0
            while (True):
                ret, img = cam.read()
                gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
                faces = detector.detectMultiScale(gray, 1.3, 5)
                for (x, y, w, h) in faces:
                    cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
                    # incrementing sample number
                    sampleNum = sampleNum + 1
```

```
        # saving the captured face in the dataset folder
        cv2.imwrite("TrainingImage/ " + Name + "." + Enrollment + '.' + str(sampleNum) + ".jpg",
                    gray)
        print("Images Saved for Enrollment :"+Enrollment)
        cv2.imshow('Frame', img)
    # wait for 100 miliseconds
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
    #
    # # break if the sample number is morethan 100
    elif sampleNum > 70:
        break



cam.release()
cv2.destroyAllWindows()
ts = time.time()
Date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
Time = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
row = [Enrollment, Name, Date, Time]
with open('StudentDetails\StudentDetails.csv', 'a+') as csvFile:
    writer = csv.writer(csvFile, delimiter=',')
    writer.writerow(row)
    csvFile.close()
res = "Images Saved for Enrollment : " + Enrollment + " Name : " + Name
```

```
            Notification.configure(
                text=res, bg="SpringGreen3", width=50, font=('times', 18, 'bold'))
            Notification.place(x=250, y=400)
        except FileExistsError as F:
            f = 'Student Data already exists'
            Notification.configure(text=f, bg="Red", width=21)
            Notification.place(x=450, y=400)



# for choose subject and fill attendance
def subjectchoose():
    def Fillattendances():
        sub = tx.get()
        now = time.time()  # For calculate seconds of video
        future = now + 20
        if time.time() < future:
            if sub == '':
                err_screen1()
            else:
                recognizer = cv2.face.LBPHFaceRecognizer_create()  # cv2.createLBPHFaceRecognizer()
                try:
                    recognizer.read("TrainingImageLabel\Trainner.yml")
                except:
                    e = 'Model not found,Please train model'
                    Notifica.configure(
```

```
    text=e, bg="red", fg="black", width=33, font=('times', 15, 'bold'))
  Notifica.place(x=20, y=250)


recognizer = cv2.face.LBPHFaceRecognizer_create()#cv2.createLBPHFaceRecognizer()
recognizer.read("TrainingImageLabel\Trainner.yml")
harcascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(harcascadePath);
df=pd.read_csv("StudentDetails\StudentDetails.csv")
cam = cv2.VideoCapture(0)
font = cv2.FONT_HERSHEY_SIMPLEX
col_names =  ['Enrollment','Name','Date','Time']
attendance = pd.DataFrame(columns = col_names)
while True:
  ret, im =cam.read()
  gray=cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
  faces=faceCascade.detectMultiScale(gray, 1.2,5)
  for(x,y,w,h) in faces:
    cv2.rectangle(im,(x,y),(x+w,y+h),(225,0,0),2)
    Id, conf = recognizer.predict(gray[y:y+h,x:x+w])
    if(conf < 70):
      global Subject
      Subject =tx.get()
      ts = time.time()
      date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
      timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
```

```
            aa=df.loc[df['Enrollment'] == Id]['Name'].values
            tt=str(Id)+"-"+aa
            attendance.loc[len(attendance)] = [Id,aa,date,timeStamp]


        else:
            Id = 'Unknown'
            tt = str(Id)
            cv2.rectangle(
                im, (x, y), (x + w, y + h), (0, 25, 255), 7)
            cv2.putText(im, str(tt), (x + h, y),
                        font, 1, (0, 25, 255), 4)
    if time.time() > future:
        break


    attendance = attendance.drop_duplicates(
        ['Enrollment'], keep='first')
    cv2.imshow('Filling attedance..', im)
    key = cv2.waitKey(30) & 0xff
    if key == 27:
        break

ts = time.time()
date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
timeStamp = datetime.datetime.fromtimestamp(
    ts).strftime('%H:%M:%S')
```

```
Hour, Minute, Second = timeStamp.split(":")
fileName = "Attendance/" + Subject + "_" + date + \
   "_" + Hour + "-" + Minute + "-" + Second + ".csv"
attendance = attendance.drop_duplicates(
   ['Enrollment'], keep='first')
print(attendance)
attendance.to_csv(fileName, index=False)

# Create table for Attendance
date_for_DB = datetime.datetime.fromtimestamp(
   ts).strftime('%Y_%m_%d')
DB_Table_name = str(
   Subject + "_" + date_for_DB + "_Time_" + Hour + "_" + Minute + "_" + Second)
import pymysql.connections

# Connect to the database
try:
   global cursor
   connection = pymysql.connect(
      host='localhost', user='root', password='', db='Face_reco_fill')
   cursor = connection.cursor()
except Exception as e:
   print(e)

sql = "CREATE TABLE " + DB_Table_name + """
```

```
(ID INT NOT NULL AUTO_INCREMENT,
 ENROLLMENT varchar(100) NOT NULL,
 NAME VARCHAR(50) NOT NULL,
 DATE VARCHAR(20) NOT NULL,
 TIME VARCHAR(20) NOT NULL,
    PRIMARY KEY (ID)
    );
"""
# Now enter attendance in Database
insert_data = "INSERT INTO " + DB_Table_name + \
   " (ID,ENROLLMENT,NAME,DATE,TIME) VALUES (0, %s, %s, %s,%s)"
VALUES = (str(Id), str(aa), str(date), str(timeStamp))
try:
   cursor.execute(sql)  # for create a table
   # For insert data into table
   cursor.execute(insert_data, VALUES)
except Exception as ex:
   print(ex)  #

M = 'Attendance filled Successfully'
Notifica.configure(text=M, bg="Green", fg="white",
                   width=33, font=('times', 15, 'bold'))
Notifica.place(x=20, y=250)

cam.release()
```

```
        cv2.destroyAllWindows()


        import csv
        import tkinter
        root = tkinter.Tk()
        root.title("Attendance of " + Subject)
        root.configure(background='grey80')
        cs = 'FaceReqG' + fileName
        with open(cs, newline="") as file:
            reader = csv.reader(file)
            r = 0


            for col in reader:
                c = 0
                for row in col:
                    # i've added some styling
                    label = tkinter.Label(root, width=10, height=1, fg="black", font=('times', 15, ' bold '),
                                          bg="white", text=row, relief=tkinter.RIDGE)
                    label.grid(row=r, column=c)
                    c += 1
                r += 1
        root.mainloop()
        print(attendance)


# windo is frame for subject chooser
```

```
windo = tk.Tk()
# windo.iconbitmap('AMS.ico')
windo.title("Enter subject name...")
windo.geometry('580x320')
windo.configure(background='grey80')
Notifica = tk.Label(windo, text="Attendance filled Successfully", bg="Green", fg="white", width=33,
                    height=2, font=('times', 15, 'bold'))


def Attf():
    import subprocess
    subprocess.Popen(
        r'explorer /select,"Attendance\"')


attf = tk.Button(windo,  text="Check Sheets", command=Attf, fg="white", bg="black",
                 width=12, height=1, activebackground="white", font=('times', 14, ' bold '))
attf.place(x=430, y=255)


sub = tk.Label(windo, text="Enter Subject : ", width=15, height=2,
               fg="black", bg="grey", font=('times', 15, ' bold '))
sub.place(x=30, y=100)


tx = tk.Entry(windo, width=20, bg="white",
              fg="black", font=('times', 23))
tx.place(x=250, y=105)
```

```
    fill_a = tk.Button(windo, text="Fill Attendance", fg="white", command=Fillattendances,
bg="SkyBlue1", width=20, height=2,
                       activebackground="white", font=('times', 15, ' bold '))
    fill_a.place(x=250, y=160)
    windo.mainloop()



def admin_panel():
    win = tk.Tk()
    # win.iconbitmap('AMS.ico')
    win.title("LogIn")
    win.geometry('880x420')
    win.configure(background='grey80')

    def log_in():
        username = un_entr.get()
        password = pw_entr.get()

        if username == 'FaceReqG':
            if password == 'FaceReqG':
                win.destroy()
                import csv
                import tkinter
                root = tkinter.Tk()
                root.title("Student Details")
                root.configure(background='grey80')
```

```
        cs = 'StudentDetails/StudentDetails.csv'
        with open(cs, newline="") as file:
            reader = csv.reader(file)
            r = 0

            for col in reader:
                c = 0
                for row in col:
                    # i've added some styling
                    label = tkinter.Label(root, width=10, height=1, fg="black", font=('times', 15, ' bold '),
                                          bg="white", text=row, relief=tkinter.RIDGE)
                    label.grid(row=r, column=c)
                    c += 1
                r += 1
        root.mainloop()
    else:
        valid = 'Incorrect ID or Password'
        Nt.configure(text=valid, bg="red", fg="white",
                     width=38, font=('times', 19, 'bold'))
        Nt.place(x=120, y=350)

else:
    valid = 'Incorrect ID or Password'
    Nt.configure(text=valid, bg="red", fg="white",
```

```
                width=38, font=('times', 19, 'bold'))
        Nt.place(x=120, y=350)

Nt = tk.Label(win, text="Attendance filled Successfully", bg="Green", fg="white", width=40,
            height=2, font=('times', 19, 'bold'))
# Nt.place(x=120, y=350)

un = tk.Label(win, text="Enter username : ", width=15, height=2, fg="black", bg="grey",
            font=('times', 15, ' bold '))
un.place(x=30, y=50)

pw = tk.Label(win, text="Enter password : ", width=15, height=2, fg="black", bg="grey",
            font=('times', 15, ' bold '))
pw.place(x=30, y=150)

def c00():
    un_entr.delete(first=0, last=22)

un_entr = tk.Entry(win, width=20, bg="white", fg="black",
                font=('times', 23))
un_entr.place(x=290, y=55)

def c11():
    pw_entr.delete(first=0, last=22)
```

```
    pw_entr = tk.Entry(win, width=20, show="*", bg="white",
                       fg="black", font=('times', 23))
    pw_entr.place(x=290, y=155)

    c0 = tk.Button(win, text="Clear", command=c00, fg="white", bg="black", width=10, height=1,
                   activebackground="white", font=('times', 15, ' bold '))
    c0.place(x=690, y=55)

    c1 = tk.Button(win, text="Clear", command=c11, fg="white", bg="black", width=10, height=1,
                   activebackground="white", font=('times', 15, ' bold '))
    c1.place(x=690, y=155)

    Login = tk.Button(win, text="LogIn", fg="black", bg="SkyBlue1", width=20,
                      height=2,
                      activebackground="Red", command=log_in, font=('times', 15, ' bold '))
    Login.place(x=290, y=250)
    win.mainloop()



# For train the model
def trainimg():
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    global detector
    detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
    try:
```

```
        global faces, Id
        faces, Id = getImagesAndLabels("TrainingImage")
    except Exception as e:
        l = 'please make "TrainingImage" folder & put Images'
        Notification.configure(text=l, bg="SpringGreen3",
                               width=50, font=('times', 18, 'bold'))
        Notification.place(x=350, y=400)

    recognizer.train(faces, np.array(Id))
    try:
        recognizer.save("TrainingImageLabel\Trainner.yml")
    except Exception as e:
        q = 'Please make "TrainingImageLabel" folder'
        Notification.configure(text=q, bg="SpringGreen3",
                               width=50, font=('times', 18, 'bold'))
        Notification.place(x=350, y=400)

    res = "Model Trained"  # +",".join(str(f) for f in Id)
    Notification.configure(text=res, bg="olive drab",
                           width=50, font=('times', 18, 'bold'))
    Notification.place(x=250, y=400)


def getImagesAndLabels(path):
    imagePaths = [os.path.join(path, f) for f in os.listdir(path)]
```

```
    # create empth face list
    faceSamples = []
    # create empty ID list
    Ids = []
    # now looping through all the image paths and loading the Ids and the images
    for imagePath in imagePaths:
        # loading the image and converting it to gray scale
        pilImage = Image.open(imagePath).convert('L')
        # Now we are converting the PIL image into numpy array
        imageNp = np.array(pilImage, 'uint8')
        # getting the Id from the image

        Id = int(os.path.split(imagePath)[-1].split(".")[1])
        # extract the face from the training image sample
        faces = detector.detectMultiScale(imageNp)
        # If a face is there then append that in the list as well as Id of it
        for (x, y, w, h) in faces:
            faceSamples.append(imageNp[y:y + h, x:x + w])
            Ids.append(Id)
    return faceSamples, Ids



window.grid_rowconfigure(0, weight=1)
window.grid_columnconfigure(0, weight=1)
# window.iconbitmap('AMS.ico')
```

```
def on_closing():
    from tkinter import messagebox
    if messagebox.askokcancel("Quit", "Do you want to quit?"):
        window.destroy()



window.protocol("WM_DELETE_WINDOW", on_closing)


message = tk.Label(window, text="Face-Recognition-Based-Attendance-Management-System",
bg="black", fg="white", width=50,
                   height=3, font=('times', 30, ' bold '))


message.place(x=80, y=20)


Notification = tk.Label(window, text="All things good", bg="Green", fg="white", width=15,
                        height=3, font=('times', 17))


lbl = tk.Label(window, text="Enter Enrollment : ", width=20, height=2,
               fg="black", bg="grey", font=('times', 15, 'bold'))
lbl.place(x=200, y=200)



def testVal(inStr, acttyp):
    if acttyp == '1':  # insert
```

```
        if not inStr.isdigit():
            return False
    return True



txt = tk.Entry(window, validate="key", width=20, bg="white",
               fg="black", font=('times', 25))
txt['validatecommand'] = (txt.register(testVal), '%P', '%d')
txt.place(x=550, y=210)

lbl2 = tk.Label(window, text="Enter Name : ", width=20, fg="black",
                bg="grey", height=2, font=('times', 15, ' bold '))
lbl2.place(x=200, y=300)

txt2 = tk.Entry(window, width=20, bg="white",
                fg="black", font=('times', 25))
txt2.place(x=550, y=310)

clearButton = tk.Button(window, text="Clear", command=clear, fg="white", bg="black",
                        width=10, height=1, activebackground="white", font=('times', 15, ' bold '))
clearButton.place(x=950, y=210)

clearButton1 = tk.Button(window, text="Clear", command=clear1, fg="white", bg="black",
                         width=10, height=1, activebackground="white", font=('times', 15, ' bold '))
clearButton1.place(x=950, y=310)
```

```
AP = tk.Button(window, text="Check Registered students", command=admin_panel, fg="black",
          bg="SkyBlue1", width=19, height=1, activebackground="white", font=('times', 15, ' bold '))
AP.place(x=990, y=410)

takeImg = tk.Button(window, text="Take Images", command=take_img, fg="black", bg="SkyBlue1",
               width=20, height=3, activebackground="white", font=('times', 15, ' bold '))
takeImg.place(x=90, y=500)

trainImg = tk.Button(window, text="Train Images", fg="black", command=trainimg, bg="SkyBlue1",
                width=20, height=3, activebackground="white", font=('times', 15, ' bold '))
trainImg.place(x=390, y=500)

FA = tk.Button(window, text="Automatic Attendance", fg="black", command=subjectchoose,
          bg="SkyBlue1", width=20, height=3, activebackground="white", font=('times', 15, ' bold '))
FA.place(x=690, y=500)

quitWindow = tk.Button(window, text="Manually Fill Attendance", command=manually_fill, fg="black",
                bg="SkyBlue1", width=20, height=3, activebackground="white", font=('times', 15, ' bold '))
quitWindow.place(x=990, y=500)

window.mainloop()
```

## Training.py

```
import cv2
import os
import numpy as np
from PIL import Image
#
# recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer = cv2.face.LBPHFaceRecognizer_create()
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")



def getImagesAndLabels(path):
    # get the path of all the files in the folder
    imagePaths = [os.path.join(path, f) for f in os.listdir(path)]
    # create empth face list
    faceSamples = []
    # create empty ID list
    Ids = []
    # now looping through all the image paths and loading the Ids and the images
    for imagePath in imagePaths:
        # loading the image and converting it to gray scale
        pilImage = Image.open(imagePath).convert('L')
        # Now we are converting the PIL image into numpy array
        imageNp = np.array(pilImage, 'uint8')
```

```
        # getting the Id from the image

        Id = int(os.path.split(imagePath)[-1].split(".")[1])
        # extract the face from the training image sample
        faces = detector.detectMultiScale(imageNp)
        # If a face is there then append that in the list as well as Id of it
        for (x, y, w, h) in faces:
            faceSamples.append(imageNp[y:y+h, x:x+w])
            Ids.append(Id)
    return faceSamples, Ids



faces, Ids = getImagesAndLabels('TrainingImage')
recognizer.train(faces, np.array(Ids))
recognizer.save('TrainingImageLabel/Trainner.yml')
```

## Testing.py

```
import cv2
import numpy as np

recognizer = cv2.createLBPHFaceRecognizer()
recognizer.read('TrainingImageLabel/Trainner.yml')
cascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath)
```

```
font = cv2.FONT_HERSHEY_SIMPLEX


cam = cv2.VideoCapture(0)
while True:
   ret, im = cam.read()
   gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
   faces = faceCascade.detectMultiScale(gray, 1.2, 5)
   for(x, y, w, h) in faces:
      Id, conf = recognizer.predict(gray[y:y+h, x:x+w])


      # # else:
      # #     Id="Unknown"
      # cv2.rectangle(im, (x-22,y-90), (x+w+22, y-22), (0,255,0), -1)
      cv2.rectangle(im, (x, y), (x + w, y + h), (0, 260, 0), 7)
      cv2.putText(im, str(Id), (x, y-40), font, 2, (255, 255, 255), 3)


      # cv2.putText(im, str(Id), (x + h, y), font, 1, (0, 260, 0), 2)
   cv2.imshow('im', im)
   if cv2.waitKey(10) & 0xFF == ord('q'):
      break
cam.release()
cv2.destroyAllWindows()
```