CODE:-

import cv2

import numpy as np

import face\_recognition

import os

import mysql.connector as sql

from datetime import datetime

from tkinter import \*

from tkinter import ttk

import tkinter as tk

import csv

con=sql.connect(host='127.0.0.1', user='root', password='', database='attendance')

cur=con.cursor(buffered=True)

def show\_frame(frame):

frame.tkraise()

cap = cv2.VideoCapture(0)

path = 'faces'

images = []

personNames = []

myList = os.listdir(path)

print(myList)

for cu\_img in myList:

current\_Img = cv2.imread(f'{path}/{cu\_img}')

images.append(current\_Img)

personNames.append(os.path.splitext(cu\_img)[0])

print(personNames)

def faceEncodings(images):

encodeList = []

for img in images:

img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

encode = face\_recognition.face\_encodings(img)[0]

encodeList.append(encode)

return encodeList

def attendance(name,time,storein):

#with open('att.csv', 'r+') as f:

# myDataList = f.readlines()

# nameList = []

# for line in myDataList:

# entry = line.split(',')

# nameList.append(entry[0])

# if name not in nameList:

# time\_now = datetime.now()

# tStr = time\_now.strftime('%H:%M:%S')

# dStr = time\_now.strftime('%d/%m/%Y')

# f.writelines(f'{name},{tStr},{dStr}\n')

cur.execute("UPDATE "+storein+" SET attendance='Present' WHERE time='"+time+"' AND regid='"+name+"'")

con.commit()

encodeListKnown = faceEncodings(images)

print('All Encodings Complete!!!')

master = Tk()

master.title('Attendance System')

master.geometry("{0}x{1}+0+0".format(master.winfo\_screenwidth(), master.winfo\_screenheight()))

master.state()

master.rowconfigure(0,weight = 1)

master.columnconfigure(0,weight = 1)

frame1 = Frame(master , bg = 'dark blue')

frame2 = Frame(master , bg ='antique white')

frame3 = Frame(master)

frame4 = Frame(master)

frame5 = Frame(master , bg ='floral white')

for frame in (frame1 , frame3,frame2,frame4,frame5):

frame.grid(row=0 , column =0 , sticky = 'snew')

#frame1

frame1.backGroundImage=PhotoImage(file="bk2.png")

frame1.backGroundImageLabel=Label(frame1,image = frame1.backGroundImage)

frame1.backGroundImageLabel.place(x=0,y=0)

a = Label(frame1 ,text="Class",font = 'Helvetica 12 bold',bg='OrangeRed2',fg='black', width = 15 , height = 2).place(relx = 0.55, rely = 0.25)

b = Label(frame1 ,text="Teacher",font = 'Helvetica 12 bold',bg='OrangeRed2',fg='black', width = 15 , height = 2).place(relx = 0.55, rely = 0.35)

c = Label(frame1 ,text="Subject",font = 'Helvetica 12 bold',bg='OrangeRed2',fg='black', width = 15 , height = 2).place(relx = 0.55, rely = 0.45)

d = Label(frame1 ,text="Month",font = 'Helvetica 12 bold',bg='OrangeRed2',fg='black', width = 15 , height = 2).place(relx = 0.55 ,rely = 0.55)

a1 = ttk.Combobox(frame1 , width = 25,height=4)

b1 = ttk.Combobox(frame1 , width = 25,height=4)

c1 = ttk.Combobox(frame1 , width = 25,height=4)

d1 = ttk.Combobox(frame1 , width = 25,height=4)

def clicked():

res = "Welcome to " + txt.get()

lbl.configure(text= res)

cur.execute("SELECT cname FROM classes")

classes=cur.fetchall()

classesl=[]

for i in classes:

classesl.append(str(i)[2:int(len(i)-4)])

a1['values'] = classesl

#a1.grid(column = 5, row =20,padx=20)

a1.place(relx=0.7,rely=0.265)

cur.execute("SELECT tname FROM teachers")

teachers=cur.fetchall()

teachersl=[]

for i in teachers:

teachersl.append(str(i)[2:int(len(i)-4)])

b1['values'] = teachersl

b1.place(relx=0.7,rely=0.365)

cur.execute("SELECT sub FROM subjects")

subs=cur.fetchall()

subsl=[]

for i in subs:

subsl.append(str(i)[2:int(len(i)-4)])

c1['values'] = subsl

c1.place(relx=0.7,rely=0.465)

cur.execute("SELECT dname FROM storein")

databases=cur.fetchall()

databasesl=[]

for i in databases:

databasesl.append(str(i)[2:int(len(i)-4)])

d1['values'] = databasesl

d1.place(relx=0.7,rely=0.565)

def sub():

cname=a1.get()

tname=b1.get()

sub=c1.get()

storein=d1.get()

cur.execute("INSERT INTO "+storein+"(rollno,regid,sname) SELECT rollno,regid,sname FROM "+cname)

con.commit()

cur.execute("SELECT time FROM "+storein+" ORDER BY id DESC LIMIT 1")

time1=cur.fetchone()

time=time1[0]

print(time)

cur.execute("UPDATE "+storein+" SET tname='"+tname+"',sub='"+sub+"',attendance='Absent' WHERE time='"+time+"'")

con.commit()

while True:

ret, frame = cap.read()

faces = cv2.resize(frame, (0, 0), None, 0.25, 0.25)

faces = cv2.cvtColor(faces, cv2.COLOR\_BGR2RGB)

facesCurrentFrame = face\_recognition.face\_locations(faces)

encodesCurrentFrame = face\_recognition.face\_encodings(faces, facesCurrentFrame)

for encodeFace, faceLoc in zip(encodesCurrentFrame, facesCurrentFrame):

matches = face\_recognition.compare\_faces(encodeListKnown, encodeFace)

faceDis = face\_recognition.face\_distance(encodeListKnown, encodeFace)

# print(faceDis)

matchIndex = np.argmin(faceDis)

if matches[matchIndex]:

name = personNames[matchIndex].upper()

# print(name)

y1, x2, y2, x1 = faceLoc

y1, x2, y2, x1 = y1 \* 4, x2 \* 4, y2 \* 4, x1 \* 4

cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)

cv2.rectangle(frame, (x1, y2 - 35), (x2, y2), (0, 255, 0), cv2.FILLED)

cv2.putText(frame, name, (x1 + 6, y2 - 6), cv2.FONT\_HERSHEY\_COMPLEX, 1, (255, 255, 255), 2)

attendance(name,time,storein)

cv2.imshow('Webcam', frame)

if cv2.waitKey(1) == 13:

show\_frame(frame5)

break

cap.release()

cv2.destroyAllWindows()

btn = Button(frame1 ,text="Submit",font = 'Helvetica 12 bold',bg='VioletRed2',fg='black', width = 13 , height = 1,command=sub).place(relx = 0.7, rely = 0.66)

btn = Button(frame1 ,text="Back",font = 'Helvetica 12 bold',bg='VioletRed2',fg='black', width = 13 , height = 1,command=lambda:show\_frame(frame5)).place(relx = 0.565, rely = 0.66)

#frame2

frame2.backGroundImage1=PhotoImage(file="pic1.png")

frame2.backGroundImageLabel1=Label(frame2,image = frame2.backGroundImage1)

frame2.backGroundImageLabel1.place(x=0,y=0)

frame2.backGroundImage2=PhotoImage(file="pic2.png")

frame2.backGroundImageLabel2=Label(frame2,image = frame2.backGroundImage2)

frame2.backGroundImageLabel2.place(x=500,y=340)

frame2\_btn=Button(frame2 , text = 'Class Wise' ,width = 20 ,bg ='goldenrod',fg='black', height = 2,font ='bold',command=lambda:show\_frame(frame3))

frame2\_btn.place(relx = 0.72, rely = 0.23)

frame2\_btn=Button(frame2 , text = 'Record Wise' ,width = 20 ,bg ='goldenrod',fg='black', height = 2,font ='bold',command=lambda:show\_frame(frame4))

frame2\_btn.place(relx = 0.1, rely = 0.7)

frame2.button=PhotoImage(file="back.png")

frame2\_btn=Button(frame2 ,image= frame2.button ,borderwidth=0,bg ='antique white',fg='black',command = lambda:show\_frame(frame5))

frame2\_btn.place(relx = 0.9, rely = 0)

#frame3

classlabel = Label(frame3 ,text = "Class:" ,width=15, height=2,font = 20).place(relx=0.01,rely=0.01)

databasel = Label(frame3 ,text = "Month:" ,width=15,height=2,font = 20).place(relx=0.25,rely=0.01)

classdd = ttk.Combobox(frame3 , width=15,height=2,font = 20)

databasedd = ttk.Combobox(frame3 , width=15,height=2,font = 20)

cur.execute("SELECT cname FROM classes")

classes=cur.fetchall()

classesl=[]

for i in classes:

classesl.append(str(i)[2:int(len(i)-4)])

classdd['values'] = classesl

classdd.place(relx=0.12,rely=0.017)

cur.execute("SELECT dname FROM storein")

databases=cur.fetchall()

databasesl=[]

for i in databases:

databasesl.append(str(i)[2:int(len(i)-4)])

databasedd['values'] = databasesl

databasedd.place(relx=0.35,rely=0.017)

def addDefaulter():

defaulter=View()

f = open('C:\Aditya\Face Recognition\defaulters.csv', 'w', newline='')

writer = csv.writer(f)

writer.writerow(['Roll-No' , 'Name' , 'Reg\_Id' , '\n'])

for i in defaulter:

writer.writerow(i)

f.close()

def View():

defaulter=[]

tree.delete(\*tree.get\_children())

cname=''

storein=''

cname=classdd.get()

storein=databasedd.get()

cur.execute("SELECT rollno,sname,regid FROM "+cname+" ")

col123 = cur.fetchall()

l1=[]

l2=[]

for i in range(len(col123)):

l2.append(col123[i-1][0])

l2.append(0)

l1.append(l2)

l2=[]

cur.execute("SELECT rollno,count(id) FROM "+storein+" where sub='CN' and attendance='Present' GROUP BY rollno ")

col4 = cur.fetchall()

l3=[]

for i in range(1,len(col123)):

for j in col4:

if i==int(j[0]):

l1[i-1][1]=j[1]

print(l1)

break

for i in l1:

l3.append(i[1])

l1=[]

l2=[]

for i in range(len(col123)):

l2.append(col123[i-1][0])

l2.append(0)

l1.append(l2)

l2=[]

cur.execute("SELECT rollno,count(id) FROM "+storein+" where sub='DWM' and attendance='Present' GROUP BY rollno ")

col5 = cur.fetchall()

l4=[]

for i in range(1,len(col123)):

for j in col5:

if i==int(j[0]):

l1[i-1][1]=j[1]

print(l1)

break

for i in l1:

l4.append(i[1])

l1=[]

l2=[]

for i in range(len(col123)):

l2.append(col123[i-1][0])

l2.append(0)

l1.append(l2)

l2=[]

cur.execute("SELECT rollno,count(id) FROM "+storein+" where sub='TCS' and attendance='Present' GROUP BY rollno ")

col6 = cur.fetchall()

l5=[]

for i in range(1,len(col123)):

for j in col6:

if i==int(j[0]):

l1[i-1][1]=j[1]

print(l1)

break

for i in l1:

l5.append(i[1])

l1=[]

l2=[]

for i in range(len(col123)):

l2.append(col123[i-1][0])

l2.append(0)

l1.append(l2)

l2=[]

print(l1)

cur.execute("SELECT rollno,count(id) FROM "+storein+" where sub='PCE' and attendance='Present' GROUP BY rollno ")

col7 = cur.fetchall()

l6=[]

for i in range(1,len(col123)):

for j in col7:

if i==int(j[0]):

l1[i-1][1]=j[1]

print(l1)

break

for i in l1:

l6.append(i[1])

l1=[]

l2=[]

for i in range(len(col123)):

l2.append(col123[i-1][0])

l2.append(0)

l1.append(l2)

l2=[]

cur.execute("SELECT rollno,count(id) FROM "+storein+" where sub='SE' and attendance='Present' GROUP BY rollno ")

col8 = cur.fetchall()

l7=[]

for i in range(1,len(col123)):

for j in col8:

if i==int(j[0]):

l1[i-1][1]=j[1]

print(l1)

break

for i in l1:

l7.append(i[1])

cur.execute("SELECT count(id) FROM "+storein+" where sub='SE' OR sub='CN' OR sub='TCS' OR sub='DWM' OR sub='PCE'GROUP BY rollno")

t=cur.fetchone()

tot=int(t[0])

for c1,c2,c3,c4,c5,c6 in zip(col123,l3,l4,l5,l6,l7):

per=((c2+c3+c4+c5+c6)/tot)\*100

formatted\_per='{0:.2f}'.format(per)

if per<50.00 :

defaulter.append(c1)

lst= list(c1)

lst.append(c2)

lst.append(c3)

lst.append(c4)

lst.append(c5)

lst.append(c6)

lst.append(formatted\_per)

tree.insert("",tk.END, values=lst)

return defaulter

tree= ttk.Treeview(frame3, column=("column1", "column2", "column3","column4","column5","column6","column7","column8","column9"), show='headings')

tree.column("#1",anchor=CENTER, stretch=NO, width=120)

tree.column("#2",anchor=CENTER, stretch=NO, width=285)

tree.column("#3",anchor=CENTER, stretch=NO, width=120)

tree.column("#4",anchor=CENTER, stretch=NO, width=120)

tree.column("#5",anchor=CENTER, stretch=NO, width=120)

tree.column("#6",anchor=CENTER, stretch=NO, width=120)

tree.column("#7",anchor=CENTER, stretch=NO, width=120)

tree.column("#8",anchor=CENTER, stretch=NO, width=120)

tree.column("#9",anchor=CENTER, stretch=NO, width=120)

tree.heading("#1", text="RollNo")

tree.heading("#2", text="Student name")

tree.heading("#3", text="regno")

tree.heading("#4", text="CN")

tree.heading("#5", text="DWM")

tree.heading("#6", text="TCS")

tree.heading("#7", text="PCEII")

tree.heading("#8", text="SE")

tree.heading("#9", text="Attendance")

tree.place(height=600,width=1235,relx=0.01,rely=0.1)

btn = Button(frame3 ,text="Search" , width = 12 , height = 1,font=20,command=View).place(relx = 0.53, rely = 0.02)

btn = Button(frame3 ,text="Back" , width = 12 , height = 1,font=20,command=lambda:show\_frame(frame2)).place(relx = 0.65, rely = 0.02)

btn = Button(frame3 ,text="Defaulters" , width = 12 , height = 1,font=20,command=addDefaulter).place(relx = 0.77, rely = 0.02)

#frame4

sname = Label(frame4 ,text="Student:" , width = 15 , height = 2).place(relx = 0.01, rely = 0.09)

regid = Label(frame4 ,text="Regid:" , width = 15 , height = 2).place(relx = 0.21, rely = 0.035)

rollno = Label(frame4 ,text="Rollno:" , width = 15 , height = 2).place(relx = 0.41, rely = 0.035)

sub = Label(frame4 ,text="Subject:" , width = 15 , height = 2).place(relx = 0.21, rely = 0.09)

month = Label(frame4 ,text="Month:" , width = 15 , height = 2).place(relx = 0.01, rely = 0.035)

teacher = Label(frame4 ,text="Teacher:" , width = 15 , height = 2).place(relx = 0.41, rely = 0.09)

monthdd = ttk.Combobox(frame4 , width = 25,height=2)

regiddd = ttk.Combobox(frame4 , width = 25)

regiddd['values']=-1

regiddd.place(relx = 0.28, rely = 0.04)

snamedd = ttk.Combobox(frame4 , width = 25)

snamedd['values']='default'

snamedd.place(relx = 0.09, rely = 0.1)

rollnodd = ttk.Combobox(frame4 , width = 25)

rollnodd['values'] =-1

rollnodd.place(relx = 0.5, rely = 0.04)

subdd = ttk.Combobox(frame4 , width = 25)

cur.execute("SELECT sub FROM subjects")

subs=cur.fetchall()

subsl=[]

for i in subs:

subsl.append(str(i)[2:int(len(i)-4)])

subdd['values']=subsl

subdd.place(relx = 0.28, rely = 0.1)

teacherdd = ttk.Combobox(frame4 , width = 25)

cur.execute("SELECT tname FROM teachers")

teachers=cur.fetchall()

teachersl=[]

for i in teachers:

teachersl.append(str(i)[2:int(len(i)-4)])

teacherdd['values']=teachersl

teacherdd.place(relx = 0.5, rely = 0.1)

cur.execute("SELECT dname FROM storein")

months=cur.fetchall()

monthsl=[]

for i in months:

monthsl.append(str(i)[2:int(len(i)-4)])

monthdd['values'] = months

monthdd.place(relx = 0.09, rely = 0.04)

def viewrec():

tree1.delete(\*tree1.get\_children())

sname=snamedd.get()

storein=monthdd.get()

regid=regiddd.get()

rollno=rollnodd.get()

sub=subdd.get()

tname=teacherdd.get()

cur.execute("SELECT \* FROM "+storein+" where regid=%s or rollno=%s or sub=%s or tname=%s or sname=%s",(regid,rollno,sub,tname,sname))

t=cur.fetchall()

for t1 in t:

tree1.insert("",tk.END, values=t1)

tree1= ttk.Treeview(frame4, column=("column1", "column2", "column3","column4","column5","column6","column7","column8"), show='headings')

tree1.column("#1",anchor=CENTER, stretch=NO, width=115)

tree1.column("#2",anchor=CENTER, stretch=NO, width=115)

tree1.column("#3",anchor=CENTER, stretch=NO, width=115)

tree1.column("#4",anchor=CENTER, stretch=NO, width=200)

tree1.column("#5",anchor=CENTER, stretch=NO, width=200)

tree1.column("#6",anchor=CENTER, stretch=NO, width=115)

tree1.column("#7",anchor=CENTER, stretch=NO, width=115)

tree1.column("#8",anchor=CENTER, stretch=NO, width=200)

tree1.heading("#1", text="Id")

tree1.heading("#2", text="RollNo")

tree1.heading("#3", text="Reg No")

tree1.heading("#4", text="Student Name")

tree1.heading("#5", text="Teacher")

tree1.heading("#6", text="Subject")

tree1.heading("#7", text="Attendance")

tree1.heading("#8", text="Time")

tree1.place(height=500,width=1200,relx=0.02,rely=0.2)

btn = Button(frame4 ,text="Search" , width = 15 , height = 2,font=23,command=viewrec).place(relx = 0.65, rely = 0.02)

btn = Button(frame4 ,text="Back" , width = 15 , height = 2,font=23,command=lambda:show\_frame(frame2)).place(relx = 0.81, rely = 0.02)

#frame5

frame5.backGroundImage=PhotoImage(file="home.png")

frame5.backGroundImageLabel=Label(frame5,image = frame5.backGroundImage)

frame5.backGroundImageLabel.place(x=0,y=0)

l=Label(frame5, text='ATTENDANCE MANAGEMENT SYSTEM', font = 'comicsansms 25 bold' , pady = 15 , padx= 20,bg='midnight blue',fg='white').place(relx = 0.3, rely = 0.1, anchor = CENTER)

frame4\_btn=Button(frame5 , text = 'Mark Attendance' ,width = 20 , height = 2,bg='steel blue',fg='black',font ='comicsansms 15 bold', command = lambda:show\_frame(frame1))

frame4\_btn.place(relx = 0.18, rely = 0.77)

frame4\_btn=Button(frame5 , text = 'View Attendance' ,width = 20 , height = 2,bg ='steel blue',fg='black',font ='comicsansms 15 bold', command = lambda:show\_frame(frame2))

master.mainloop()



OUTPUT:-











