ABSTRACT

- Currently, schools and other organizations are using personal identification strategies such as facial recognition for the identification of people and the result will be taken as attendance. When attendance is done manually it generates a lot of waste on time productivity. So we a proposed solution for this problem is through automation of the attendance system using face recognition. The face is the primary identification for any human. Hence this project can be used for many other applications where face recognition can be used for authentication. This project describes the method of detecting and recognizing the face in real time with the help of Raspberry Pi. Raspberry Pi along with a Camera module makes the whole process simpler. Raspberry Pi is also used for processing the images captured through Camera module V2 using Open CV and it transfers the data through the internet. This system will automatically update the student's presence in the class to the student's database and sends the message to guardians of absentees and also to the Head of the department. This process is done without any human intervention.

INDEX

S.NO	CONTENTS	PAGE NO
1	INTRODUCTION	04
2	BACKGROUND	05
3	PROBLEM DEFINITION	06
4	OBJECTIVES	07
5	METHODOLOGY/PROCEDURE	08-12
6	RESULTS AND DISCUSSION	13
7	CONCLUSION AND FUTURE SCOPE	14
8	REFERENCES	15
9	CODES IN APPENDIX	16-19

INTRODUCTION

- Face detection is one of the current research topics in the computer vision field. It is a simple task for computers to detect faces and store them in their database.
- The facial recognition attendance system uses facial recognition technology to identify an individual using facial features and marks their attendance automatically. It helps identify and recognize a person digitally. The algorithm compares a person's facial features against the facial features of the stored images in the database. The convenience factor here is better and more straightforward when compared to the regular way.
- The software can be used for different groups of people such as employees, students, etc. The system records and stores the data in real time.
- Unless other biometrics, the facial recognition system is a touchless way to manage students and management. It helps to manage the inflow and outflow of people in schools and offices safely and efficiently.

BACKGROUND

- Face recognition has been invented more than 50 years ago**.**
- In 1967, **Woodrow W. Bledsoe**, a pioneer in artificial intelligence, developed a system that classified photos of faces through a graphical computer input device called a RAND tablet.
- A research team led by **Woodrow W Bledsoe** ran experiments between 1964 and 1966 to see whether computers can recognize human faces. The team used a rudimentary scanner to map the person's hairline, eyes, and nose. The task of the computer was to find the matches. It was unsuccessful.
- Thanks to camera technology improvements, mapping processes, machine learning, and processing speeds, facial recognition has come of age**.**
- Most systems use 2D camera technology, which creates a flat image of a face, and maps 'nodal points' (size/shape of eyes, nose, cheekbones, etc.). The system then calculates the nodes' relative position and converts the data into a numerical code. The recognition algorithms search a stored database of faces for a match.
- As the days passed, people started to use facial recognition for attendance.

PROBLEM DEFINITION

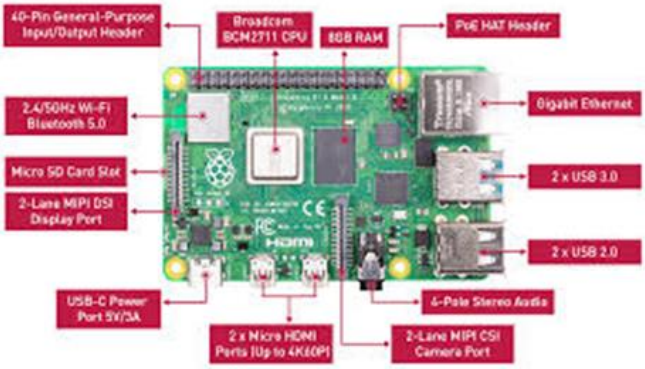
- PROBLEM STATEMENT To develop an automated attendance system using face recognition.
- Concept is in a classroom with a large number of students, which is a very tedious and time-consuming task to take attendance manually. Therefore, we can implement an effective system that will mark the attendance of students automatically by recognizing their faces. The process of this face recognition system is divided into various steps, but the important steps are the detection of the face and recognition of the face. Firstly, to mark the attendance of students, the image of a person's face will be required. This image can be snapped from the camera device, which will be placed in the classroom at a suitable location from where the whole classroom can be covered. This image will act as input to the system. For effective face detection, the image needs to be enhanced by using some image processing techniques like grayscale conversion of the image and histogram equalization. To identify the state students sitting on the last rows neatly, the histogram equalization of the image needs to be done. Hence, there is a need to develop a real-time operating student attendance system which means the identification process must be done within defined time constraints to prevent omission. The extracted features from facial images which represent the identity of the students have to be consistent towards a change in background, illumination, pose, and expression. High accuracy and fast computation time will be the evaluation points of the performance.

OBJECTIVES

- To design a face recognition-based attendance system for school events.
- The system should serve as a platform wherein the student's attendance during school events will be monitored.
- The system is simple to use and understand for the officer in charge.
- The system should reduce the number of errors in keeping track of students' attendance.
- The system should simplify monitoring and reduce the amount of time spent on it.

METHODOLOGY/PROCEDURE

- There are five major steps followed to develop attendance with facial recognition using Raspberry pi. Raspberry pi implementation, VNC server, Installing PhpMyAdmin, creating a database, and testing the database.
- **Raspberry pi implementation**
- The raspberry pi 3B + contains a 2GHZ and 5GHZ IEEE. It is connected t the computer with an ethernet cable. The network that is connected is 10/100 MBPS in wireless LAN.
- The Raspberry pi is connected to a camera module with the help of another slot area**.**



Installing the dependencies for

The following libraries are required to run this project.

- OpenCV

  OpenCV is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage. The library is cross-platform and free for use under the open-source Apache 2 License.

- NumPy

  NumPy is a library for the Python programming language, that adds support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

- MySQL Connector

  MySQL is a relational database management system (RDBMS) developed by Oracle that is based on structured query language (SQL). A database is a structured collection of data.

- **Installation of VNC**

  VNC server is an information source for programming functions mainly aimed at real-time computer vision.

  First, we need to download the VNC server from an access network, and then we need to enter the credentials to show up to the source.

  To compile the code we use the Thonny compiler, which will be available at the advanced options at the top.

  The code will be stored in the file database in advanced options.

- **Creating Database**

  As the facial method has been chosen for implementation, it is crucial for the acceptance of every individual whose attendance needs to be taken.

  Here the face of every individual needs to be captured and stored in PhpMyAdmin.

  PhpMyAdmin is a free and open-source administration tool for MySQL and MariaDB.

  Credentials are given in PhpMyAdmin, and should also be converted into Open CV code.

  We use XAMPP a free open source, for starting modifiers such as Apache and MySQL.

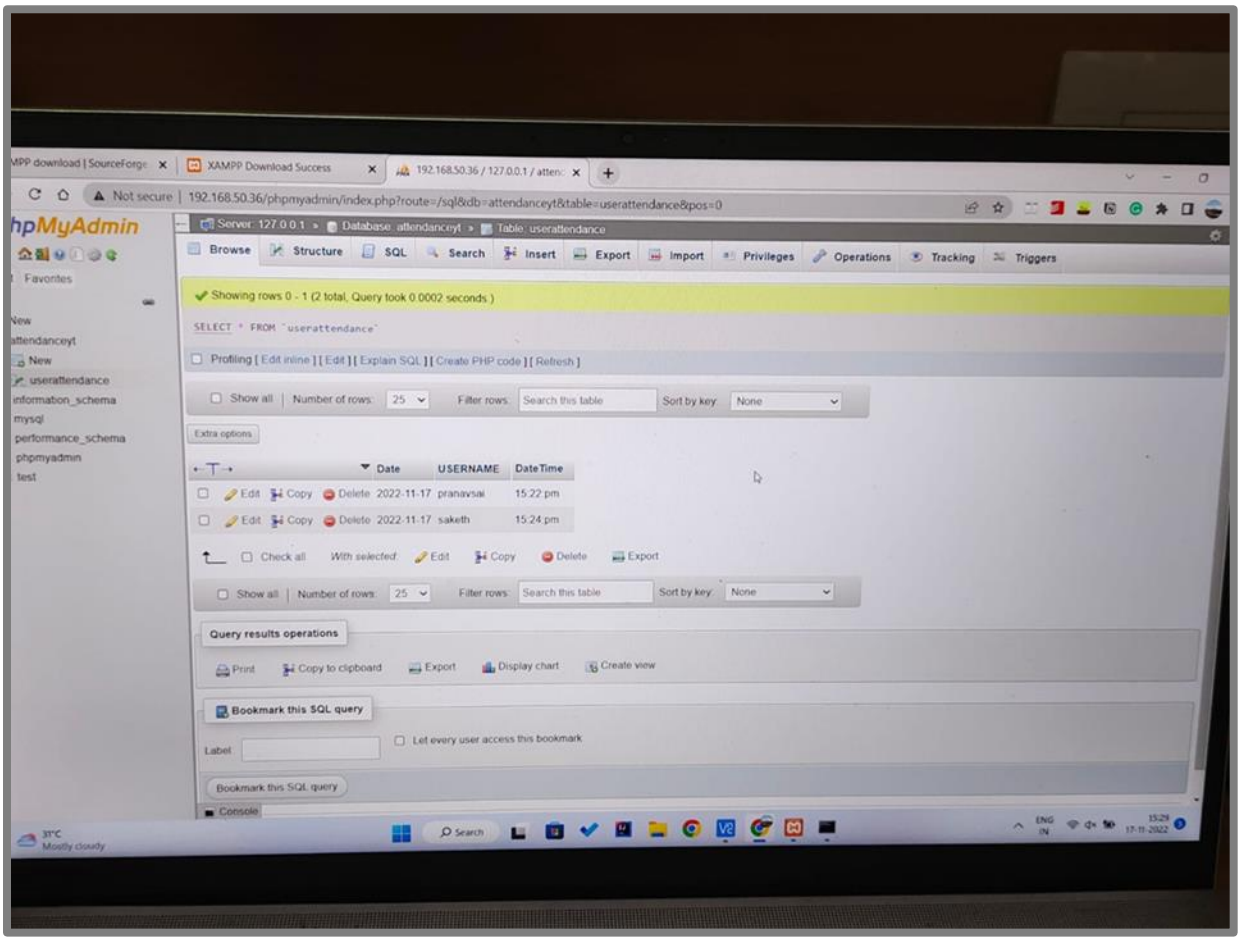
  Here multiple samples of pictures are taken by students. All their profiles are stored in the database of PhpMyAdmin.

- **Testing Database**

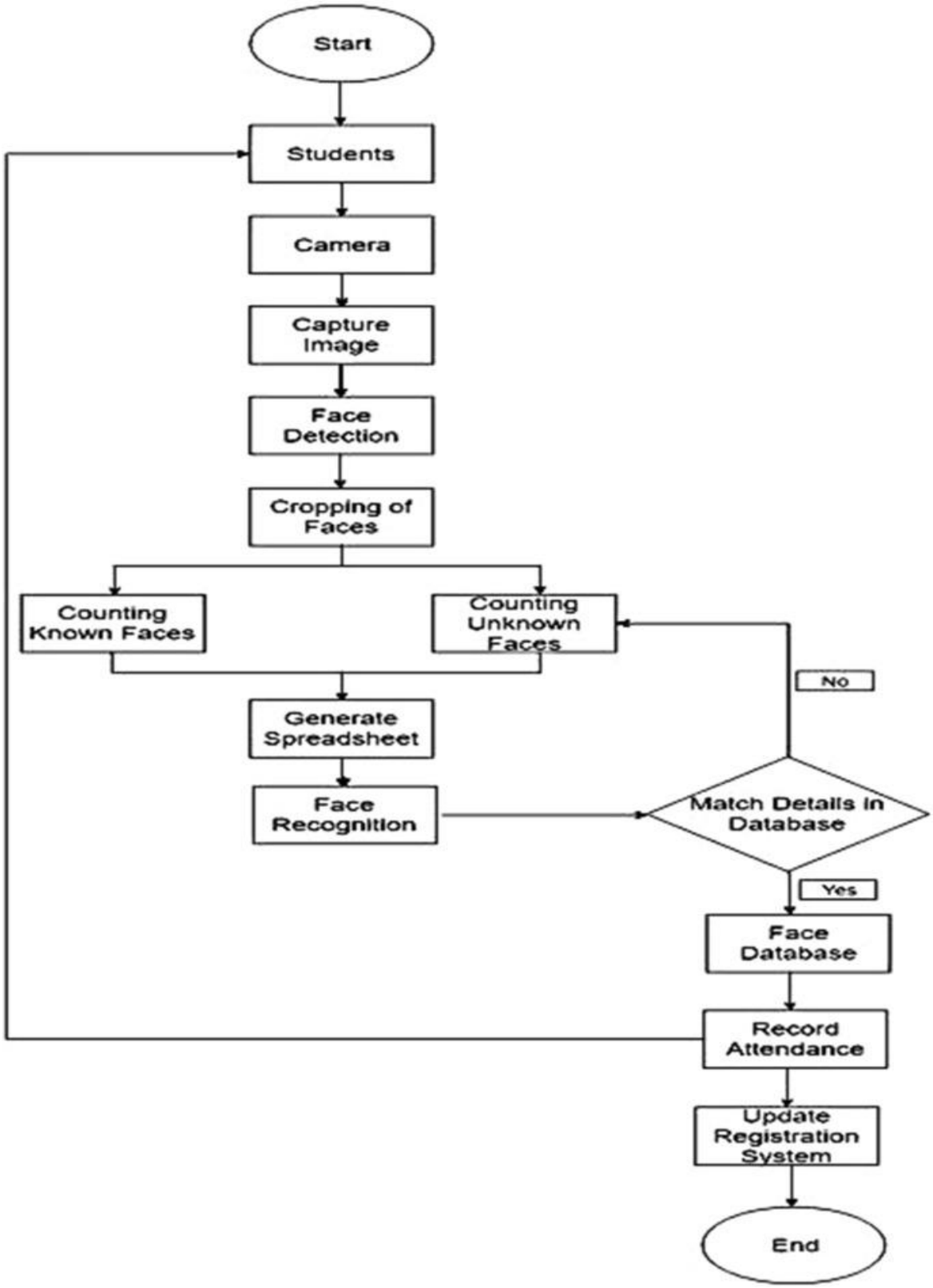
  The ability of this algorithm to recognize faces that lies on how well it can be extracted and classify the faces. Since an image may contain unnecessary background and elements other than faces it is important to remove those complications.

  Whenever the camera module detects an image, the algorithm compares it with the pictures that are stored in the database.

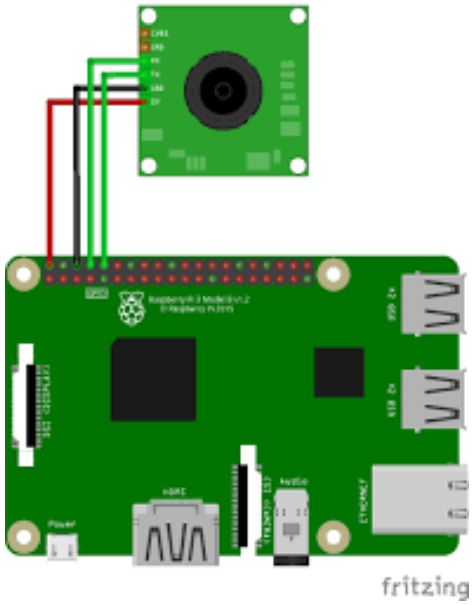
  When the match is detected, then the name is represented under the exposed picture. The data will be automatically registered in PhpMyAdmin.



Flow chart :-

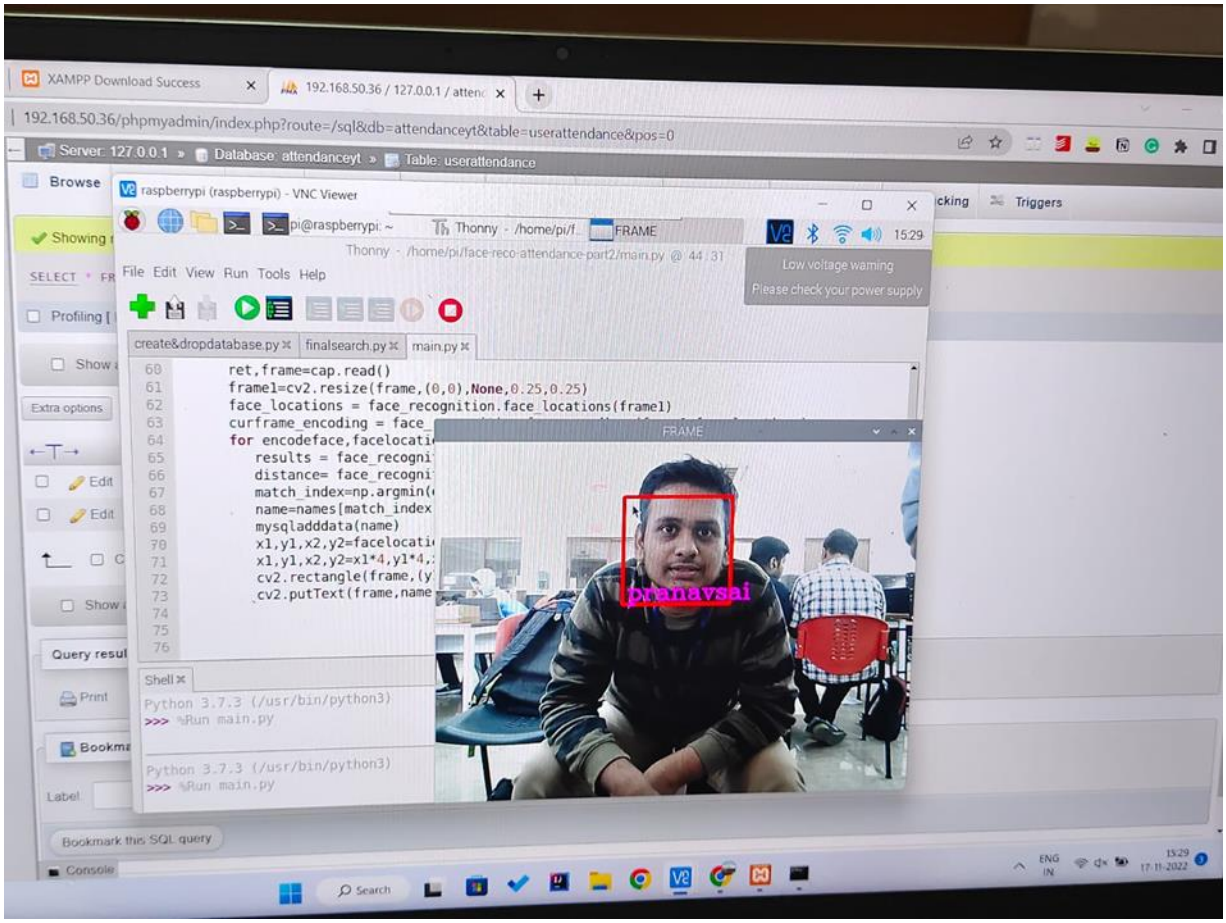

Start
Students
Camera
Capture Image
Face Detection
Cropping of Faces
Counting Known Faces
Counting Unknown Faces
Generate Spreadsheet
Face Recognition
Match Details in Database
No
Yes
Face Database
Record Attendance
Update Registration System
End

CIRCUIT DIAGRAM:



RESULT AND DISCUSSION

We created an output where the student images will be automatically stored in the database of raspberry pi 4B. The figure underneath will explain how the output of the system will detect an individual's picture when the match is recognized.



The square indicates the face of the presence of the person's face and detects his name as stored in the database.

PROBLEMS FACED:

- A network connection is one of the major reasons for raspberry pi to suddenly detect an error. At any moment when the network changes, it leads to a change of IP address. This may cause an error in the code since we use the previous IP address for execution.
- Another major problem is communication with a computer with raspberry pi. Since the raspberry pi we used is a 2 GB model it caused us a delay in implementing the code without any errors.

CONCLUSION AND FUTURE SCOPE

- This project can be applied to several security applications where authentication is needed to access the individual of their respective system.
- Face recognition can be used in their recognizing guilty parties involved in unauthorized business.
- Face recognition can be improved concerning the utilization of resources so that it can recognize more faces at a time which can make the system better.
- We came to realize that there are an extensive variety of methods to recover facial data which are non-productive.
- Hence we accomplished building up a solid and productive participation framework to actualize an image-handling algorithm to identify faces in the classrooms or other respective organizations to precisely digitally check their attendance.

Applications

- Schools
- Universities
- Offices
- Health Care
- At often in politics.

REFERENCES :-

- Robu.in(2021). Face recognition using raspberry pi | Real-time Face Recognition. https://www.youtube.com/watch?v=9OOYgWjWehI
- Freedom Tech(2022). Face recognition attendance system | OpenCV python| Computer vision | raspberry pi OpenCV. https://www.youtube.com/watch?v=-T6797cfPuk
- Facial recognition system- Wikipedia https://en.wikipedia.org/wiki/Facial_recognition_system
- Facial recognition attendance system for Human Resources https://www.timedynamo.com/blog/face-recognition-attendance-system
- OpenCV course full tutorial https://youtu.be/oXlwWbU8l2o
- Github.com https://github.com/jasmcaus/opencv-co...
- Xampp Server https://en.wikipedia.org/wiki/XAMPP
- Create a database and table in MySQL with phpMyAdmin https://www.youtube.com/watch?v=j8KUiRPGSPs

CODES IN APPENDIX

Main.py

```
import cv2
import glob
import face_recognition
import numpy as np
from datetime import datetime
from datetime import date
import mysql.connector
import os



FONT=cv2.FONT_HERSHEY_COMPLEX
images=[]
names=[]
today = date.today()
now = datetime.now()
dtString = now.strftime("%H:%M:%P")
path = "/home/pi/images/."
for file in glob.glob(path):
    image = cv2.imread(file)
    a=os.path.basename(file)
    b=os.path.splitext(a)[0]
    names.append(b)
    images.append(image)



def encoding1(images):
    encode=[]

    for img in images:
        unk_encoding = face_recognition.face_encodings(img)[0]
        encode.append(unk_encoding)
    return encode

encodelist=encoding1(images)
def mysqladddata(names):
        mydb = mysql.connector.connect(
        host= "192.168.27.36",
        user="projectyt",
        password="Pranavsai@2003" ,
        database="Attendanceyt"
```

```
)

      a = mydb.cursor()
      sql = ("INSERT IGNORE INTO UserAttendance(Date,USERNAME,DATETIME)
VALUE(%s,%s,%s)")
      data=(today,names,dtString)

      a.execute(sql,data)


      mydb.commit()
      mydb.close()

cap =cv2.VideoCapture(0)
while True:
     ret,frame=cap.read()
     frame1=cv2.resize(frame,(0,0),None,0.25,0.25)
     face_locations = face_recognition.face_locations(frame1)
     curframe_encoding = face_recognition.face_encodings(frame1,face_locations)
     for encodeface,facelocation in zip(curframe_encoding,face_locations):
       results = face_recognition.compare_faces(encodelist, encodeface)
       distance= face_recognition.face_distance(encodelist, encodeface)
       match_index=np.argmin(distance)
       name=names[match_index]
       mysqladddata(name)
       x1,y1,x2,y2=facelocation
       x1,y1,x2,y2=x1*4,y1*4,x2*4,y2*4
       cv2.rectangle(frame,(y1,x1),(y2,x2),(0,0,255),3)
       cv2.putText(frame,name,(y2+6,x2-6),cv2.FONT_HERSHEY_COMPLEX,1,(255,0,255),2)



     cv2.imshow("FRAME",frame)
     if cv2.waitKey(1)&0xFF==27:
        break
cap.release()
cv2.destroyAllWindows()
```

Create&Dropdatabase.py

```
import mysql.connector



    mydb = mysql.connector.connect(
        host= "192.168.27.36",
        user="projectyt",
        password="Pranavsai@2003"
    #       database="Attendanceyt"



    )
    a = mydb.cursor()
    #a.execute("CREATE DATABASE Attendanceyt")
    #a.execute("CREATE TABLE UserAttendance (Date VARCHAR(30),USERNAME
VARCHAR(255),DateTime VARCHAR(255), PRIMARY KEY (Date,USERNAME))")
    sql = "DROP DATABASE Attendanceyt"
    a.execute(sql)
    mydb.commit()
    mydb.close()
```

Final Search.py

```
import mysql.connector
```

```
def mysqlsearch(name):
    mydb = mysql.connector.connect(
    host= "192.168.27.36",
    user="projectyt",
    password="Pranavsai@2003"
    database="Attendanceyt"


)


    a = mydb.cursor()
    sql1=a.execute("SELECT * from UserAttendance WHERE USERNAME = %(name)s",
{'name': name})


    a.execute(sql1)
    checkUsername = a.fetchall()
    print(*checkUsername, sep = "\n")


    mydb.commit()
    mydb.close()


name=input("Enter UserName:-")
mysqlsearch(name).
```