A
Mini-Project Report on

FACIAL RECOGNITION ATTENDANCE SYSTEM USING PYTHON AND OPENCV

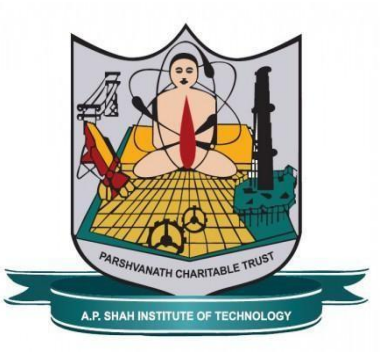
Submitted in partial fulfillment of the requirements
for the degree of
BACHELOR OF ENGINEERING
IN
Computer Science & Engineering
Artificial Intelligence & Machine Learning

by

Shashikant Shukla	21106024
Sachin Sapkale	21106026
Vikas Pandit	21106044
Jayesh Patil	21106048

Under the guidance of

Dr. Jaya Gupta



Department of Computer Science & Engineering
(Artificial Intelligence & Machine Learning)
A. P. Shah Institute of Technology
G. B. Road, Kasarvadavali, Thane (W)-400615
University Of Mumbai

2022-2023

A. P. SHA H INSTITUTE OF TECHNOLOGY

CERTIFICATE

This is to certify that the project entitled **"Facial recognition attendance system using python & opencv"** is a bonafide work of "Shashikant Shukla(21106024), Sachin Sapkale(21106026), Vikas Pandit(21106044), Jayesh Patil(21106048)"submitted to the University of Mumbai in partial fulfillment of the requirement for the award of **Bachelor of Engineering** in **Computer Science & Engineering (Artificial Intelligence & Machine Learning).**

_____	_____
Dr. Jaya Gupta	Dr. Jaya Gupta
Mini Project Guide	Head of Department

A. P. SHA H INSTITUTE OF TECHNOLOGY

Project Report Approval

This Mini project report entitled "**Facial recognition attendance system using python & opencv"** by **Shashikant Shukla, Sachin Sapkale, Vikas Pandit, Jayesh Patil** is approved for the degree of ***Bachelor of Engineering*** in ***Computer Science &Engineering***, (AIML) ***2022-23***.

External Examiner: _____

Internal Examiner: _____

Place: APSIT, Thane
Date:

Declaration

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission hasnot been taken when needed.

Shashikant Shukla	Sachin Sapkale	Vikas Pandit	Jayesh Patil
(21106024)	(21106026)	(21106044)	(21106048)

ABSTRACT

Facial recognition technology is rapidly gaining popularity due to its advanced capabilities in identifying individuals in various applications such as security systems, social media platforms, and advertising. This project proposes the use of facial recognition technology in attendance management systems to provide efficient record-keeping, streamlined processes, and accurate tracking.

The proposed Facial Recognition Attendance System utilizes Python and OpenCV for face recognition and attendance management. The system utilizes a default list feature to identify individuals and record their attendance automatically. The OpenCV computer vision library detects facial features and performs real-time face recognition. The system is highly scalable and can be customized to meet the specific needs of educational institutions, corporate offices, and other organizations.

The Facial Recognition Attendance System is a revolutionary solution that can enhance traditional attendance management methods by providing efficient attendance tracking, improving student and employee performance, and enhancing the overall management of educational institutions and organizations. The project aims to demonstrate the potential of facial recognition technology in creating a more secure and efficient workplace.

Keywords: Facial Recognition, Attendance System, Python, OpenCV, Computer Vision, Default List.

Index

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

In recent years, facial recognition technology has emerged as a reliable and efficient method for identifying individuals. The technology has numerous applications in various fields, including security, law enforcement, and access control. One of the most promising applications of facial recognition technology is attendance management systems. The use of such systems can simplify attendance taking, eliminate the need for manual attendance marking, and reduce the risk of errors.

The objective of this project is to design and develop a facial recognition attendance system using Python and OpenCV.The system is designed to automatically identify students or employees using facial recognition technology and mark their attendance. The proposed system uses the OpenCV library for face detection and recognition and Python programming language for implementing the system.

The system comprises two main parts: the face recognition algorithm and the attendance marking module. The face recognition algorithm is responsible for detecting and recognizing faces in the images captured by the camera. The attendance marking module is responsible for marking the attendance of the recognized individuals.

The face recognition algorithm is implemented using the OpenCV library, which provides a pre-trained model for face detection and recognition. The algorithm uses Haar cascades for face detection and the Local Binary Patterns Histograms (LBPH) algorithm for face recognition. The LBPH algorithm is a powerful and efficient method for facial recognition, which compares the local texture of an image with pre-stored templates to identify a face.

The attendance marking module is implemented using Python, which provides a user-friendly interface for attendance management. The module uses a database to store the attendance records of the recognized individuals. The system is designed to automatically mark the attendance of individuals whose faces are recognized. The attendance records are stored in a database, which can be easily accessed and managed by the administrator.

The proposed system has several advantages over traditional attendance management systems. The system eliminates the need for manual attendance marking, which reduces the risk of errors and saves time. The system can also generate automatic attendance reports, which can be used for various purposes, including payroll management, performance evaluation, and attendance

tracking. The system is also highly accurate and reliable, as it uses facial recognition technology, which is known for its high accuracy and reliability.

In conclusion, the facial recognition attendance system developed using Python and OpenCV is a highly efficient and reliable system for attendance management. The system can simplify attendance taking, eliminate the need for manual attendance marking, and reduce the risk of errors. The system is highly accurate and reliable, as it uses facial recognition technology for identifying individuals. The system can be easily integrated into various institutions, including schools, colleges, and offices, to manage attendance efficiently.

1.1 Objective

The main objective of this project is to automate the attendance marking process using facial recognition technology. The system aims to reduce the manual effort required for attendance marking and provide a more accurate and efficient way to mark attendance.

1.2 Scope

The main intention of this project is to solve the issues encountered in the old attendance system while reproducing a brand new innovative smart system that can provide convenience to the institution. In this project, an application will be developed which is capable of recognizing the identity of each individuals and eventually record down the data into a database system. Apart from that, an excel sheet is created which shows the students attendance.

CHAPTER 2

LITERATURE SURVEY

2. LITERATURE SURVEY

2.1-HISTORY

Facial recognition technology has come a long way since its early days of 2D image analysis. Today's facial recognition systems use complex algorithms that analyze various facial features such as the distance between the eyes, the shape of the jawline, and the curvature of the lips to create a unique "faceprint" for each individual. This faceprint is then compared to a database of known faces to determine a match

The accuracy of facial recognition systems has greatly improved over the years, thanks in large part to advances in artificial intelligence and machine learning. These systems are now able to detect faces even under challenging conditions such as poor lighting or facial occlusions (such as wearing glasses or a hat).

In recent years, facial recognition technology has become increasingly popular for use in attendance systems. Traditionally, attendance systems have relied on manual methods such as taking attendance sheets or using RFID tags to track attendance. However, these methods are often time-consuming and prone to errors. Facial recognition attendance systems offer a more efficient and accurate solution.

Facial recognition attendance systems work by using a camera to capture an image of a person's face. The system then analyzes the facial features in the image to create a faceprint, which is compared to a database of known faces to determine a match. If a match is found, the system records the individual's attendance.

Python and OpenCV have emerged as popular tools for developing facial recognition attendance systems due to their ease of use and flexibility. OpenCV provides a range of tools and algorithms for image and video processing, including facial detection and recognition. Python is a high-level programming language that is easy to learn and use, making it an ideal choice for developers.

To develop a facial recognition attendance system using Python and OpenCV, developers typically start by training the system on a set of known faces. This involves capturing multiple

images of each individual from different angles and under different lighting conditions. The system then uses these images to create a faceprint for each individual.

Once the system has been trained, it can be used to identify individuals in real-time. When a person enters the room, the system captures an image of their face and compares it to the database of known faces. If a match is found, the individual's attendance is recorded

Facial recognition attendance systems offer a number of advantages over traditional attendance systems. They are faster and more accurate, and they eliminate the need for manual tracking methods such as attendance sheets or RFID tags. Additionally, they can be integrated with other systems such as access control or time and attendance systems, making them a powerful tool for organizations looking to streamline their attendance tracking processes.

2.2-LITERATURE REVIEW

1. Face Recognition Attendance System Based on Real-Time Video Processing by Hao Yang1 and Xiaofeng Han 2 (2020)

In this paper, the authors present a face recognition attendance system that utilizes real-time video processing. The system is designed to provide an efficient and accurate method for tracking attendance in various settings such as schools, universities, and workplaces.

The system utilizes Python and OpenCV to process video streams in real-time, allowing it to quickly and accurately identify individuals based on their facial features. The algorithm used in the system is based on Local Binary Pattern Histograms (LBPH), a widely-used method for facial recognition that is known for its accuracy and efficiency.

The authors conducted experiments to test the system's accuracy and performance, and the results showed that it was able to correctly identify individuals with a high degree of accuracy, even in varying lighting conditions and with different facial expressions.

The system also includes features such as a user interface for managing attendance records and a notification system for informing users of attendance status. The authors note that the system is scalable and can be easily customized to meet the needs of different organizations and

settings.[1]

Overall, the face recognition attendance system presented in this paper offers a promising solution for accurately and efficiently tracking attendance using real-time video processing. The use of Python and OpenCV, along with the LBPH algorithm, make this system an effective tool for organizations seeking to automate their attendance tracking process

2. Facial Recognition Attendance System Using Python and OpenCv by Dr. V Suresh, Srinivasa Chakravarthi Dumpa, Chiranjeevi Deepak Vankayala, HaneeshaAduri, Jayasree Rapa, (2019)

This system uses Python and OpenCV for facial recognition attendance tracking. It captures images using a camera and processes them with OpenCV. The facial recognition algorithm identifies individuals and tracks attendance. It improves accuracy and efficiency while reducing manual labor. Features include report generation and attendance record viewing.[2]

3. Python Based Attendance System by Dr. J. Ravindranadh1 , Peram Venkata Ravindra Reddy2 (2020)

They discuss the advantages and limitations of these systems and highlight the importance of using appropriate algorithms and reliable databases. They also identify some challenges associated with the use of face recognition technology, such as privacy concerns and the need for careful camera placement and lighting conditions.

The authors suggest that integrating the attendance system with other technologies can enhance its functionality and usefulness. They conclude that a well-designed and implemented system can provide a practical solution for improving attendance management in various settings, while also addressing privacy concerns and ensuring data security. Overall, their literature review provides a useful overview of the current state of research in this field, highlighting both the potential benefits and challenges of using face recognition technology for attendance tracking.[3]

4. Automated Attendance System Using OpenCV Naman Gupta, Purushottam Sharma, Vikas Deep, Vinod Kumar Shukla (2020)

The automated attendance system using OpenCV technology creates a unique "faceprint" for each individual and compares it to a database of known faces, capturing attendance automatically. A GUI allows for real-time viewing of attendance data and adjustments.

The system has a high degree of accuracy, can be customized, and is reliable with a low error rate and fast response time. It offers a practical solution for attendance management in various settings, including classrooms, lecture halls, offices, and other workplaces.[4]

5. Review of Face Recognition Techniques by Kamini Solanki , Prashant Pittalia

The review paper by Kamini Solanki and Prashant Pittalia discusses face recognition techniques, including their advantages and limitations. The authors first provide an overview of the history of face recognition and then delve into various approaches, including Eigenface, Fisherface, Local Binary Patterns (LBP), and Convolutional Neural Networks (CNNs).

They highlight the strengths and weaknesses of each approach and discuss their applications in areas such as security systems, biometric identification, and video surveillance. The authors also explore the challenges of face recognition, including illumination, pose, expression, and occlusion, and how these challenges are being addressed.

Overall, the paper provides a comprehensive review of face recognition techniques and their potential application.[5]

2.3-LITERATURE TABLE

Sr No	Existing system	Features	Benefits	Limitation
1.	Face Recognition Attendance System Based on Real-Time Video Processing , Hao Yang1 and Xiaofeng Han2,(2020)	1)Generating reports 2)Viewing attendance records	easily accessible	Masked faces were not recognized
2.	Facial Recognition Attendance System Using Python and OpenCV, Dr. V Suresh, Srinivasa Chakravarthi Dumpa, Chiranjeevi Deepak Vankayala, Haneesha Adhuri, Jayasree Rapa,(2019)	1)Face detection 2)Face recognition	Multiple face detection was possible	Success rate is only 82%
3.	Python Based Attendance System, Dr. J Ravindranadh1, Peram Venkata Ravindra Reddy2,(2020)	1)Database Management 2)Attendance Management	Multiple faces can be detected at a time	Accuracy is low only 50% faces were recognized
4.	Automated Attendance System Using OpenCv, Naman Gupta, Purushottam Sharma, Vikas Deep, Vinod Kumar Shukla,(2020)	1)Real-time Monitoring 2) Use Eigen faces for Recognition	High accuracy	Multiple faces were not recognized.
5.	Review of Face Recognition Techniques, Kamini Solanki, Prashant Pittalia,(2020)	1)Security 2)Scalability	Used for security purposes in organizations	Doesn't recognize properly in poor light

CHAPTER 3

Problem Statement

3. Problem Statement

To develop an automated attendance system using face recognition. Concept In a classroom with large number of students, it is a very tedious and time-consuming task to take the attendance manually. Therefore, we can implement an effective system which 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 detection of face and recognition of face.

CHAPTER 4

Experimental Setup

4. Experimental Setup

4.1 Hardware Setup

Processor	: Intel Pentium III or later
Main Memory(RAM)	: 256 MB
Cache Memory	: 512 KB

4.2 Software Setup

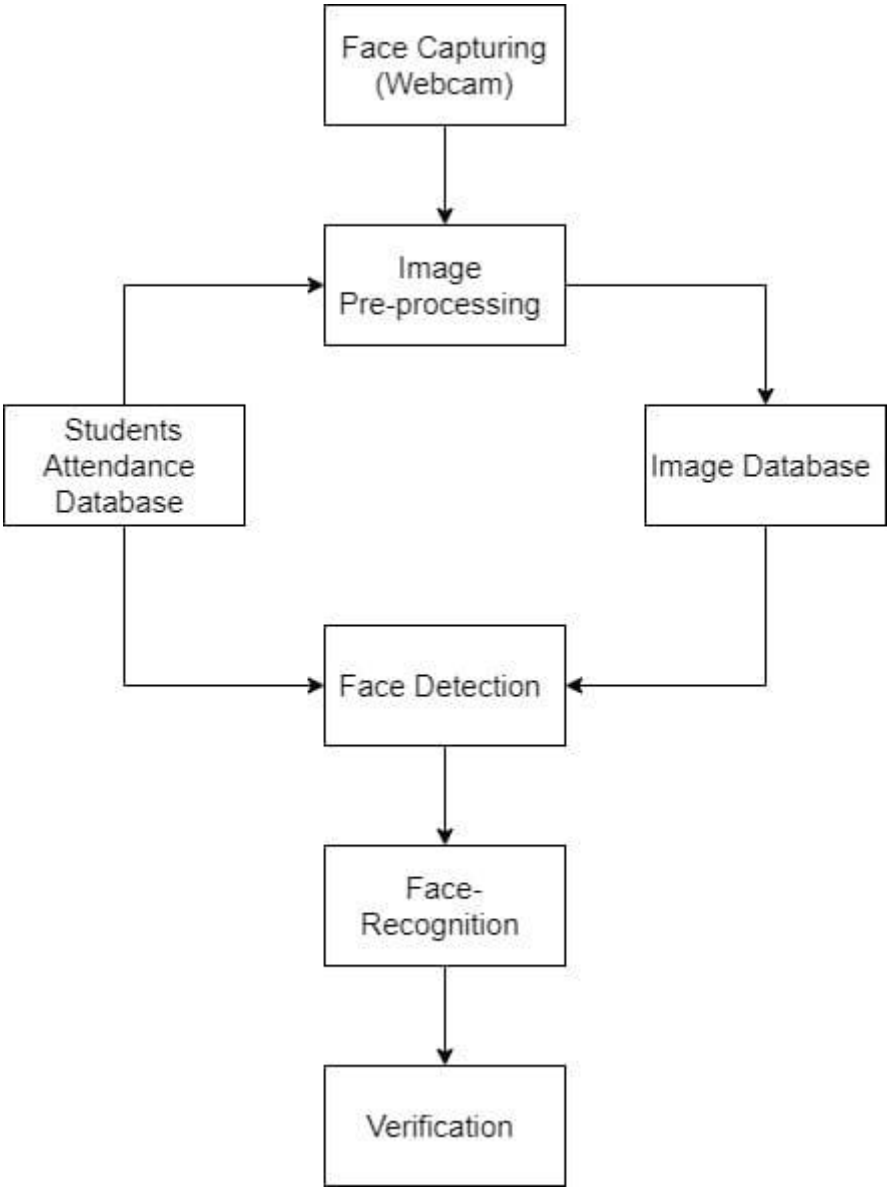
Python
Visual Studio Code
Firebase Database.

CHAPTER 5

Proposed System & Implementation

5.Proposed system & Implementation

5.1 Block diagram of proposed system



. Figure-5.1.1 Block Diagram

5.2 Description of block diagram

The block diagram starts with a block labeled "Face Capturing" which refers to the process of capturing an image of a person's face using a camera or other imaging device.

The captured image is then passed to the "Image Processing" block, which refers to a set of algorithms and techniques used to enhance, analyze, and manipulate the image data.

The "Image Detection" block then analyzes the processed image to identify the presence of a face using face detection algorithms. If a face is detected, the system proceeds to the next block labeled "Student Database."

The "Student Database" block contains a database of student records and associated images. The system compares the detected face with the images in the student database to determine if there is a match.

If there is a match, the system proceeds to the "Face Recognition" block, which refers to a set of algorithms used to identify a person's unique facial features and characteristics.

Finally, the system performs "Verification" to confirm the identity of the person in the image, and if the verification is successful, it may grant access to a specific area or perform some other action based on the application of the system.

5.3Implementation

1)Code

1.Main.py

```
import os
import pickle
import numpy as np
import cv2
import face_recognition
import cvzone
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db
from firebase_admin import storage
import numpy as np
from datetime import datetime
import pandas as pd

cred = credentials.Certificate("serviceAccountKey.json")
firebase_admin.initialize_app(cred, {
    'databaseURL': "https://faceattendancerealtime-4d6a5-default-
rtdb.firebaseio.com/",
```

```
    'storageBucket': "faceattendancerealtime-4d6a5.appspot.com"
})


bucket = storage.bucket()
# attendance = pd.read_csv("attendance.csv")

cap = cv2.VideoCapture(0)
cap.set(3, 640)
cap.set(4, 480)

imgBackground = cv2.imread('Resources/background.png')

folderModePath = 'Resources/Modes'
modelPathList = os.listdir(folderModePath)
imgModeList = []
for path in modelPathList:
    imgModeList.append(cv2.imread(os.path.join(folderModePath,path)))
    # img = cv2.imread(os.path.join(folderModePath, path))
    # print(len(imgModeList))

print("Loading Encode File ...")
file = open('EncodeFile.p', 'rb')
encodeListKnownWithIds = pickle.load(file)
file.close()
encodeListKnown, studentIds = encodeListKnownWithIds
# print(studentIds)
print("Encode File Loaded")

modeType = 0
counter = 0
id = -1
imgStudents = []

while True:
    success, img = cap.read()

    imgS = cv2.resize(img, (0, 0), None, 0.25, 0.25)
    imgS = cv2.cvtColor(imgS, cv2.COLOR_BGR2RGB)

    faceCurFrame = face_recognition.face_locations(imgS)
    encodeCurFrame = face_recognition.face_encodings(imgS, faceCurFrame)


    imgBackground[162:162 + 480, 55:55 + 640] = img
    imgBackground[44:44 + 633, 808:808 + 414] = imgModeList[modeType]

    if faceCurFrame:
        for encodeFace, faceLoc in zip(encodeCurFrame,faceCurFrame):
```

```
            matches = face_recognition.compare_faces(encodeListKnown,
encodeFace)
            faceDis = face_recognition.face_distance(encodeListKnown,
encodeFace)
            print("matches", matches)
            print("faceDis", faceDis)



            matchIndex = np.argmin(faceDis)
            # print("Match Index", matchIndex)

            if matches[matchIndex]:
            # print("Known Face Detected")
            # print(studentIds[matchIndex])
                y1, x2, y2, x1 = faceLoc
                y1, x2, y2, x1 = y1*4, x2*4, y2*4, x1*4
                bbox =55+x1, 162+y1, x2 - x1, y2-y1
                imgBackground = cvzone.cornerRect(imgBackground, bbox,
rt=0)
                id = studentIds[matchIndex]
                if counter == 0:
                    cvzone.putTextRect(imgBackground,"Loading",(275, 400))
                    cv2.imshow("Face Attendance", imgBackground)
                    cv2.waitKey(1)
                    counter = 1
                    modeType = 1

        if counter!=0:

            if counter==1:
                studentInfo = db.reference(f'Students/{id}').get()
                print(studentInfo)
                blob = bucket.get_blob(f'Images/{id}.png')
                array = np.frombuffer(blob.download_as_string(), np.uint8)
                imgStudents = cv2.imdecode(array,cv2.COLOR_BGRA2BGR)
                datetimeObject =
datetime.strptime(studentInfo['last_attendance_time'],
                                                  "%Y-%m-%d %H:%M:%S")
                secondElapsed= (datetime.now()-
datetimeObject).total_seconds()
                print(secondElapsed)
                if secondElapsed >=30:
                    ref = db.reference(f'Students/{id}')
                    studentInfo['total_attendance'] +=1
                    ref.child('total_attendance').set(studentInfo['total_at
tendance'])
                    ref.child('last_attendance_time').set(datetime.now().st
rftime("%Y-%m-%d %H:%M:%S"))
```

```
                else:
                    modeType = 3
                    counter = 0
                    imgBackground[44:44 + 633, 808:808 + 414] =
imgModeList[modeType]

            if modeType!= 3:
                if 10<counter<20:
                    modeType = 2
                imgBackground[44:44 + 633, 808:808 + 414] =
imgModeList[modeType]


   if counter<=10:
                    cv2.putText(imgBackground,
str(studentInfo['total_attendance']),(861,125),
                                cv2.FONT_HERSHEY_COMPLEX,1,(255, 255,
255),1)
                    cv2.putText(imgBackground, str(studentInfo['major']),
(1006, 550),
                                cv2.FONT_HERSHEY_COMPLEX, 0.5, (255, 255,
255), 1)
                    cv2.putText(imgBackground, str(id), (1006, 493),
                                cv2.FONT_HERSHEY_COMPLEX, 0.5, (255, 255,
255), 1)
                    cv2.putText(imgBackground,str(studentInfo['standing']),
(910,625),
                                cv2.FONT_HERSHEY_COMPLEX,0.6,(100, 100,
100),1)
                    cv2.putText(imgBackground,str(studentInfo['year']),(102
5,625),
                                cv2.FONT_HERSHEY_COMPLEX,0.6,(100, 100,
100),1)
                    cv2.putText(imgBackground,str(studentInfo['starting_yea
r']),(1124,625),
                                cv2.FONT_HERSHEY_COMPLEX,0.6,(100, 100,
100),1)

                    (w,h), _ =
cv2.getTextSize(studentInfo['name'],cv2.FONT_HERSHEY_COMPLEX, 1,1)
                    offset = (414-w)//2
                    cv2.putText(imgBackground,str(studentInfo['name']),(808
+offset, 445),
                                cv2.FONT_HERSHEY_COMPLEX,1,(50, 50,
50),1)

                    imgBackground[175:175+216,909:909+216] =
imgStudents
```

```
                counter+=1

                if counter>=20:
                    counter = 0
                    modeType = 0
                    studentInfo = []
                    imgStudent = []
                    imgBackground[44:44 + 633, 808:808 + 414] =
imgModeList[modeType]
    else:
        modeType = 0
        counter = 0
  # cv2.imshow("Face Attendance", img)
    cv2.imshow("Face Attendance" , imgBackground)
    cv2.waitKey(1)


#    # create a pandas dataframe to hold the attendance records
#     df = pd.DataFrame(attendance.items(), columns=['Name', 'Attendance'])

#    # create an Excel writer object
#     writer = pd.ExcelWriter('attendance.xlsx', engine='xlsxwriter')

# # write the dataframe to the Excel sheet
#     df.to_excel(writer, sheet_name='Attendance', index=False)

# # save the Excel sheet
#     writer.save()

#     print("Attendance has been saved to attendance.xlsx")
```

2)Encoding Generator.py

```
import cv2
import face_recognition
import pickle
import os
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db
from firebase_admin import  storage

cred = credentials.Certificate("serviceAccountKey.json")
firebase_admin.initialize_app(cred, {
    'databaseURL': "https://faceattendancerealtime-4d6a5-default-
rtdb.firebaseio.com/",
    'storageBucket': "faceattendancerealtime-4d6a5.appspot.com"
})
```

```
# Importing student images
folderPath = 'Images'
pathList = os.listdir(folderPath)
print(pathList)
imgList = []
studentIds = []
for path in pathList:
    imgList.append(cv2.imread(os.path.join(folderPath, path)))
    studentIds.append(os.path.splitext(path)[0])
    # print(path)
    # print(os.path.splitext(path)[0])

    fileName = f'{folderPath}/{path}'
    bucket = storage.bucket()
    blob = bucket.blob(fileName)
    blob.upload_from_filename(fileName)


print(studentIds)

def findEncodings(imageList):
    encodeList = []
    for img in imageList:
        img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)[0] # Modify this line
to use face locations
        encodeList.append(encode)

    return encodeList

print("Encoding Started.....")
encodeListKnown = findEncodings(imgList)
encodeListKnownWithIds = [encodeListKnown, studentIds]
print("Encoding complete")

file = open("EncodeFile.p", 'wb')
pickle.dump(encodeListKnownWithIds,file)
file.close()
```

3) Add data to database.py

```
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db

cred = credentials.Certificate("serviceAccountKey.json")
firebase_admin.initialize_app(cred, {
```

```
    'databaseURL': "https://faceattendancerealtime-4d6a5-default-
rtdb.firebaseio.com/"
})

ref = db.reference('Students')

data = {
    "21106026":
        {
            "name": "Sachin Sapkale",
            "major": "CSE(AIML)",
            "starting_year": 2022,
            "total_attendance": 0,
            "standing": "G",
            "year": 4,
            "last_attendance_time": "2022-12-11 00:54:34"
        },
    "21106048":
        {
            "name": "Jayesh Patil",
            "major": "CSE(AIML)",
            "starting_year": 2022,
            "total_attendance": 0,

             "standing": "G",
            "year": 4,

            "last_attendance_time": "2022-12-11 00:54:34"
        },
    "211060100":
        {
            "name": "Pavan Sapkale",
            "major": "CSE(AIML)",
            "starting_year": 2022,
            "total_attendance": 0,
            "standing": "G",
            "year": 4,
            "last_attendance_time": "2022-12-11 00:54:34"
        },
    "211060200":
        {
            "name": "Narendra Modi",
            "major": "CSE(AIML)",
            "starting_year": 2022,
            "total_attendance": 0,
            "standing": "G",
            "year": 4,
            "last_attendance_time": "2022-12-11 00:54:34"
        },
```

```
  "21106015":
      {
          "name": "Adrian Thereparambil",
          "major": "CSE(AIML)",
          "starting_year": 2022,
          "total_attendance": 0,
          "standing": "G",
          "year": 4,
          "last_attendance_time": "2022-12-11 00:54:34"
      },
      "21106027":
      {
          "name": "Mihir Manjrekar",
          "major": "CSE(AIML)",
          "starting_year": 2022,
          "total_attendance": 0,
          "standing": "G",
          "year": 4,
          "last_attendance_time": "2022-12-11 00:54:34"
      },
      "21106056":
      {
          "name": "Sanjita Shukla",
          "major": "CSE(AIML)",
          "starting_year": 2022,
          "total_attendance": 0,
          "standing": "G",

          "year": 4,
          "last_attendance_time": "2022-12-11 00:54:34"
      },

         "21106058":

{

"name": "Chirag Sawant",
          "major": "CSE(AIML)",
          "starting_year": 2022,
          "total_attendance": 0,
          "standing": "G",
          "year": 4,
          "last_attendance_time": "2022-12-11 00:54:34"
      },
      "21106030":
      {
          "name": "Shubham Singh",
          "major": "CSE(AIML)",
          "starting_year": 2022,
```

```
            "total_attendance": 0,
            "standing": "G",
            "year": 4,
            "last_attendance_time": "2022-12-11 00:54:34"
        },
        "21106024":
        {
            "name": "Shashikant Shukla",
            "major": "CSE(AIML)",
            "starting_year": 2022,
            "total_attendance": 0,
            "standing": "G",
            "year": 4,
            "last_attendance_time": "2022-12-11 00:54:34"
        },
        "21106035":
        {
            "name": "Vinaykumar Yadav",
            "major": "CSE(AIML)",
            "starting_year": 2022,
            "total_attendance": 0,
            "standing": "G",
            "year": 4,
            "last_attendance_time": "2022-12-11 00:54:34"
        }


}

for key, value in data.items():
    ref.child(key).set(value)
```

4) Defaulters.py

```
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db
import pandas as pd

# Initialize Firebase app
cred = credentials.Certificate("serviceAccountKey.json")
firebase_admin.initialize_app(cred, {
    'databaseURL': "https://faceattendancerealtime-4d6a5-default-rtdb.firebaseio.com/"
})

# Read student attendance data from Firebase
ref = db.reference('Students')
students_data = ref.get()

# Convert student attendance data to pandas dataframe
```

```
students_df = pd.DataFrame.from_dict(students_data, orient='index')

# Calculate attendance percentage for each student
students_df['attendance_percentage'] = students_df['total_attendance'] / 15 * 100

# Define attendance threshold for defaulters
attendance_threshold = 60

# Identify defaulters based on attendance threshold
defaulters_df = students_df.loc[students_df['attendance_percentage'] <
attendance_threshold]

# Convert defaulters data to Excel sheet
defaulters_df.to_excel('defaulters.xlsx')
```

2) Screenshot

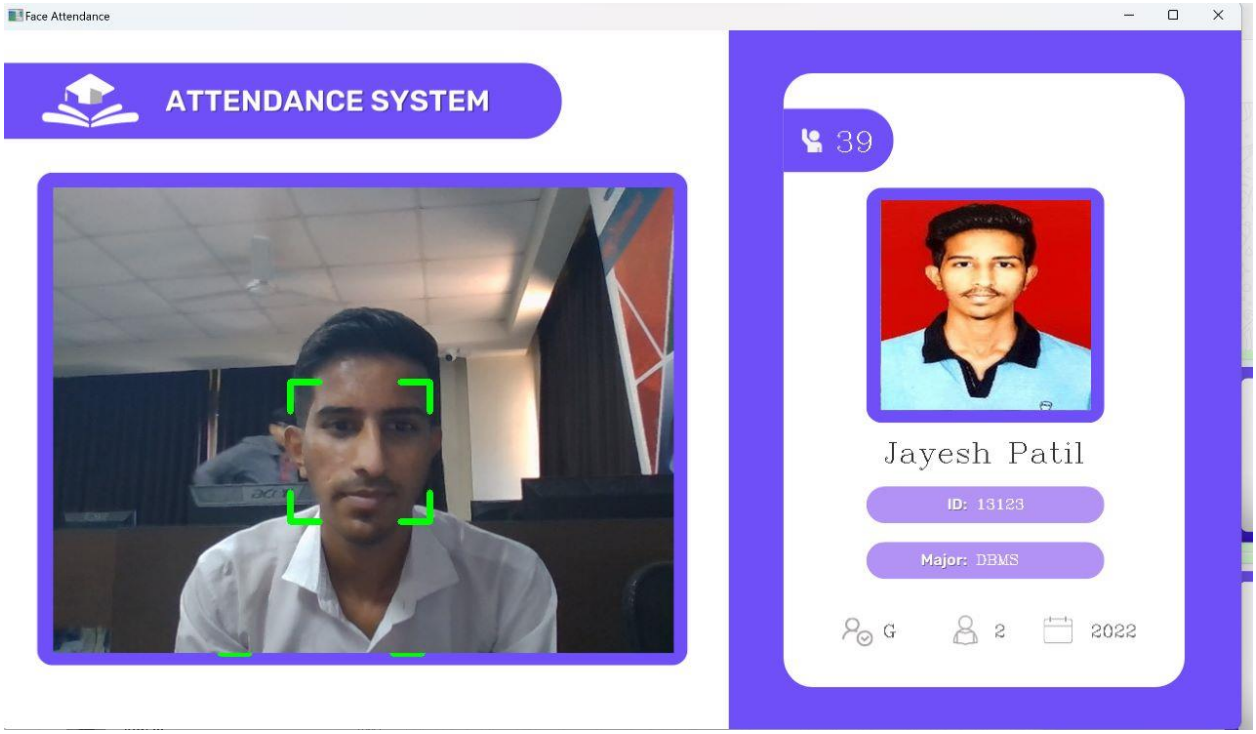


Figure-5.3.2.1 Taking attendance

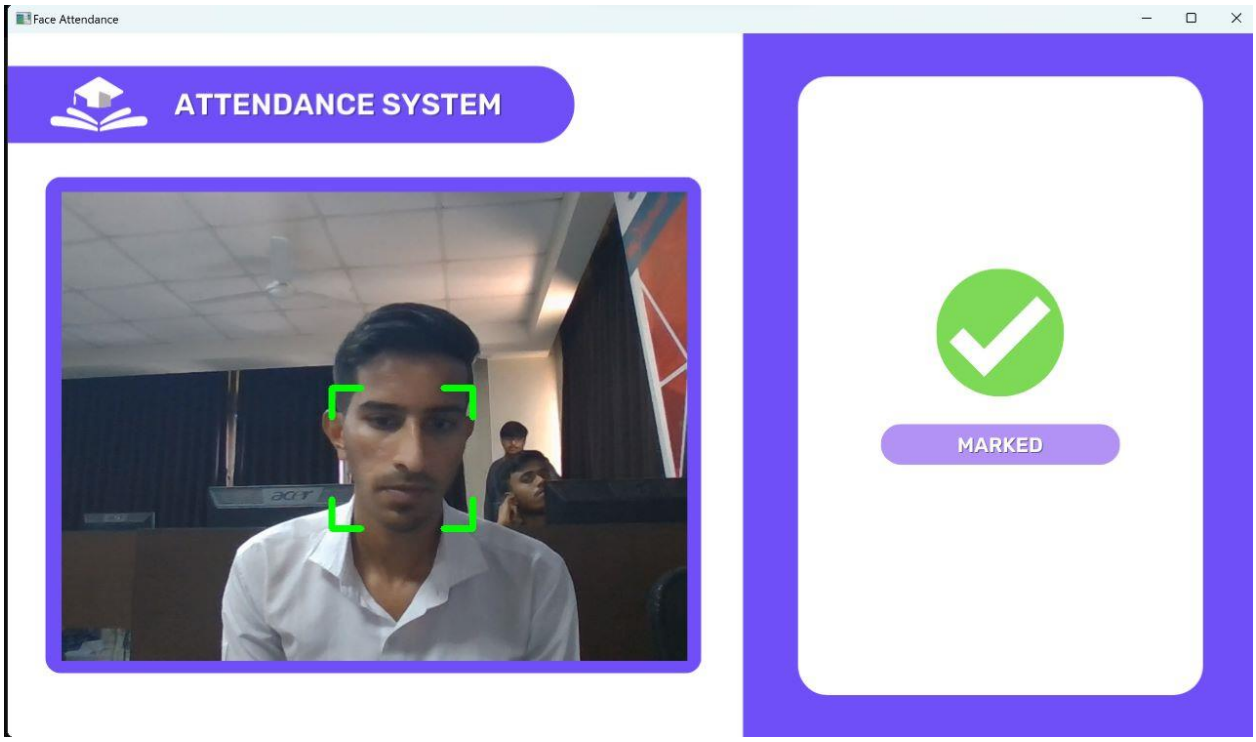


Figure-5.3.2.2 Marked attendance

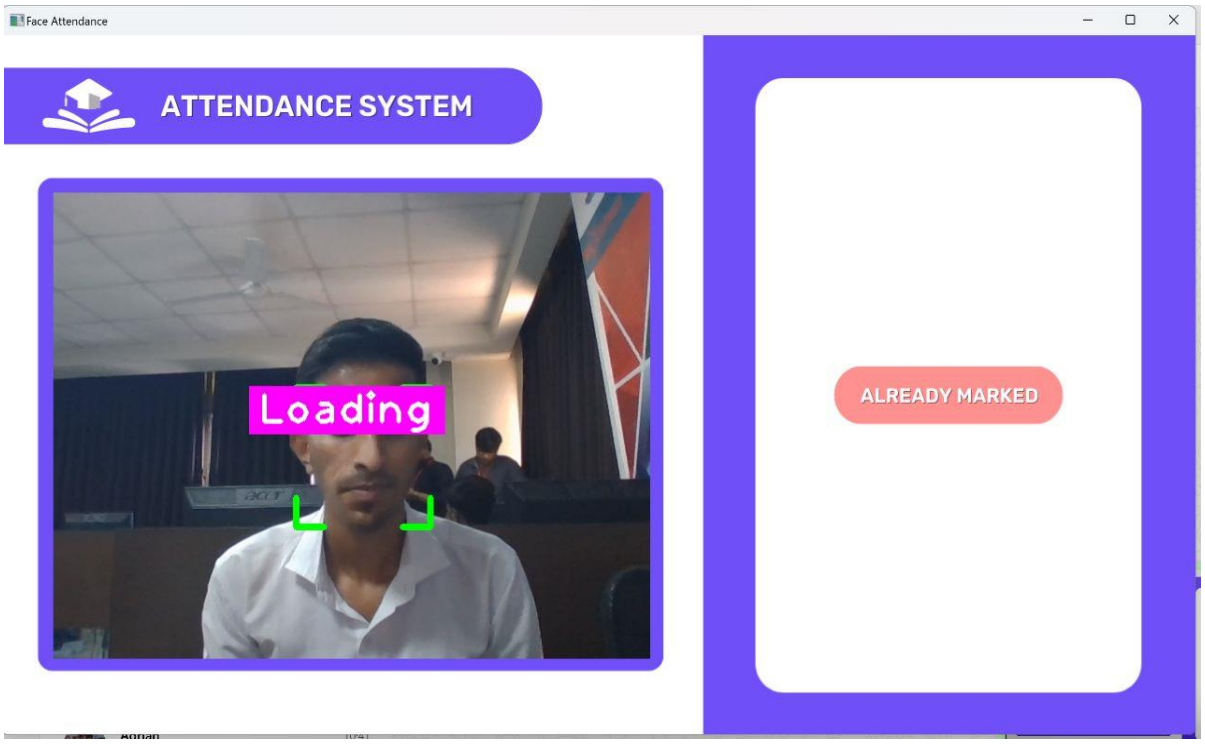


Figure-5.3.2.3 Already marked

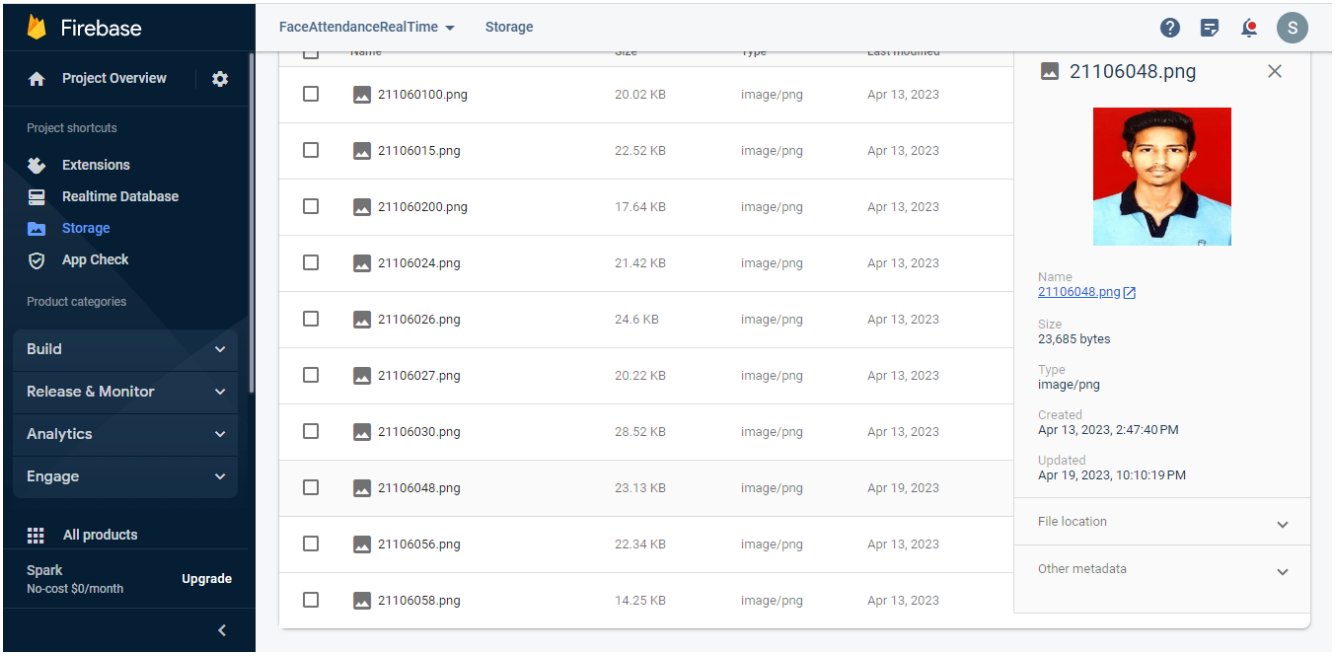


Figure-5.3.2.4 Images in Realtime database

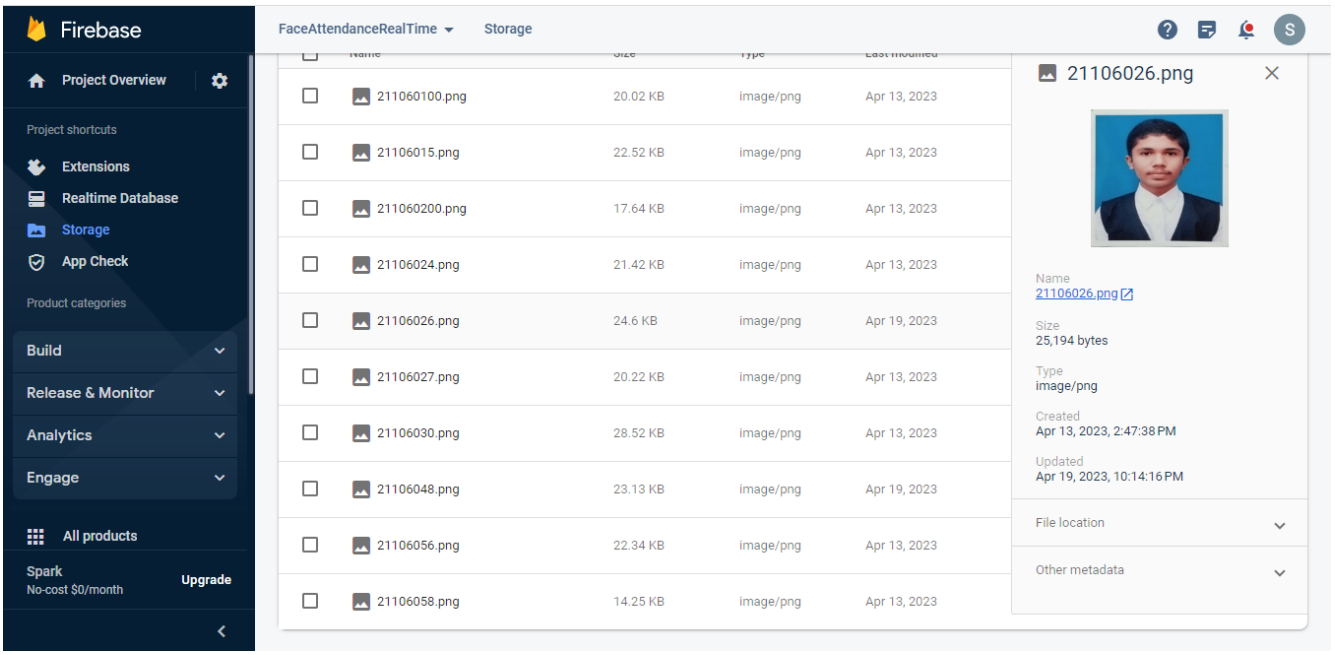


Figure-5.3.2.5 Images in Realtime database

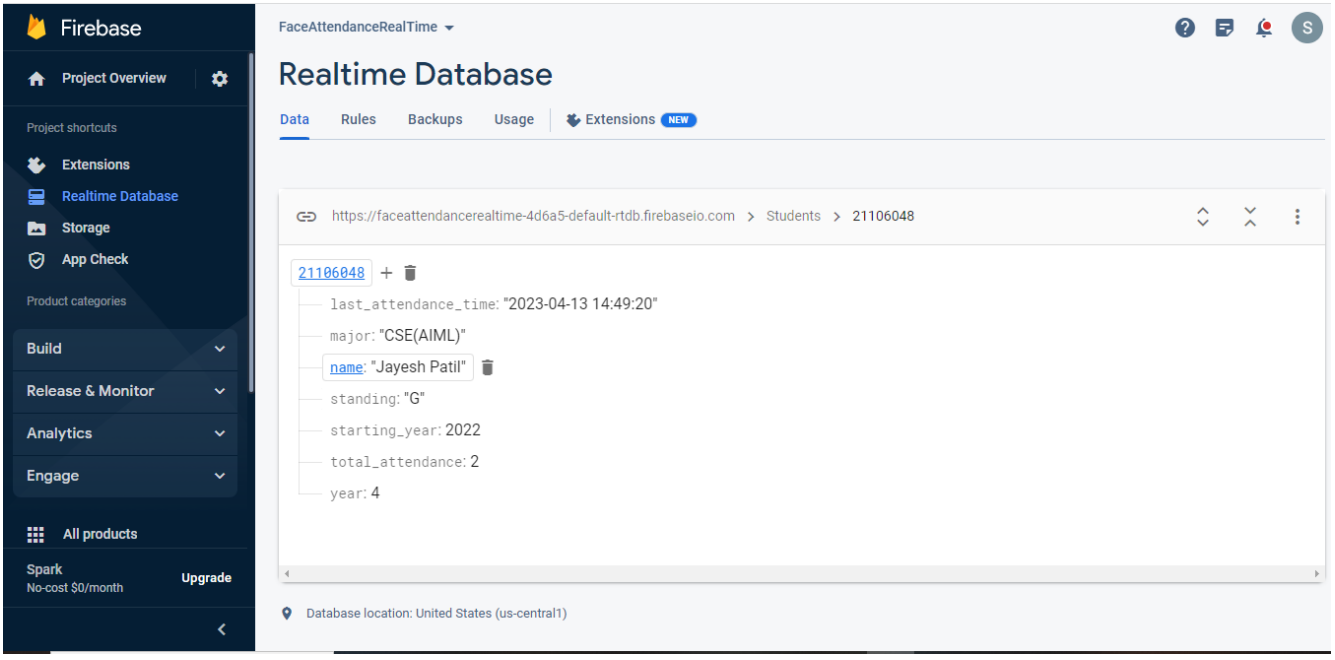


Figure-5.3.2.6 Students Attendance is Dataabse

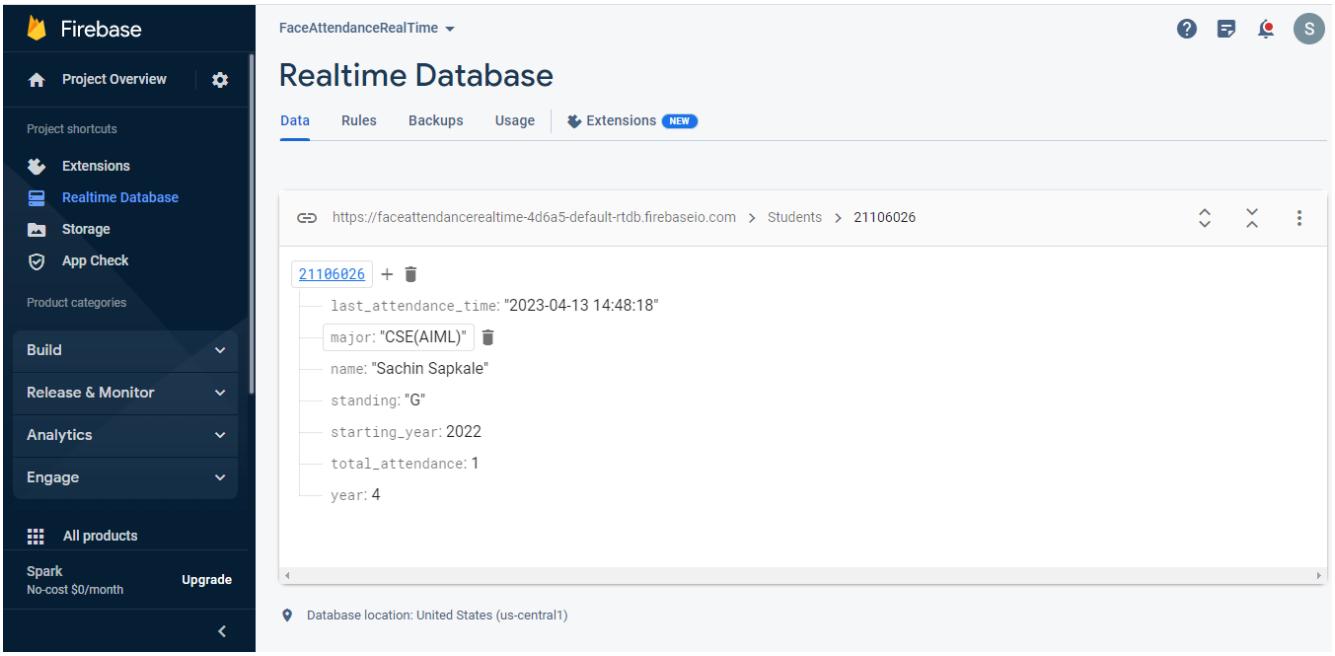
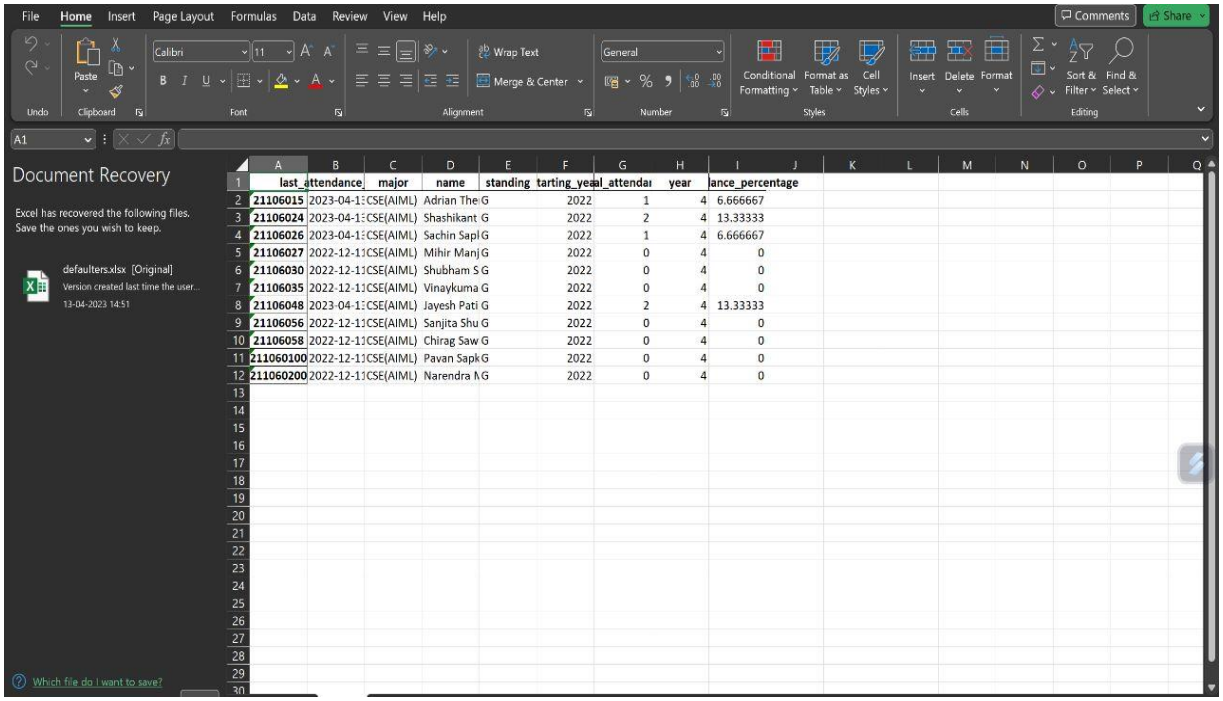


Figure-5.3.2.7 Students Attendance is Dataabse

Figure-5.3.2.7 Images



	last_	attendance_	major	name	standing	tarting_yea	al_attendar	year	ance_percentage
2	21106015	2023-04-1	CSE(AIML)	Adrian The	G	2022	1	4	6.666667
3	21106024	2023-04-1	CSE(AIML)	Shashikant	G	2022	2	4	13.33333
4	21106026	2023-04-1	CSE(AIML)	Sachin Sapl	G	2022	1	4	6.666667
5	21106027	2022-12-1	CSE(AIML)	Mihir Manj	G	2022	0	4	0
6	21106030	2022-12-1	CSE(AIML)	Shubham S	G	2022	0	4	0
7	21106035	2022-12-1	CSE(AIML)	Vinaykuma	G	2022	0	4	0
8	21106048	2023-04-1	CSE(AIML)	Jayesh Pati	G	2022	2	4	13.33333
9	21106056	2022-12-1	CSE(AIML)	Sanjita Shu	G	2022	0	4	0
10	21106058	2022-12-1	CSE(AIML)	Chirag Saw	G	2022	0	4	0
11	211060100	2022-12-1	CSE(AIML)	Pavan Sapk	G	2022	0	4	0
12	211060200	2022-12-1	CSE(AIML)	Narendra M	G	2022	0	4	0

Figure-5.3.2.8 Defaulter List

5.4 Data flow diagram

Level 0

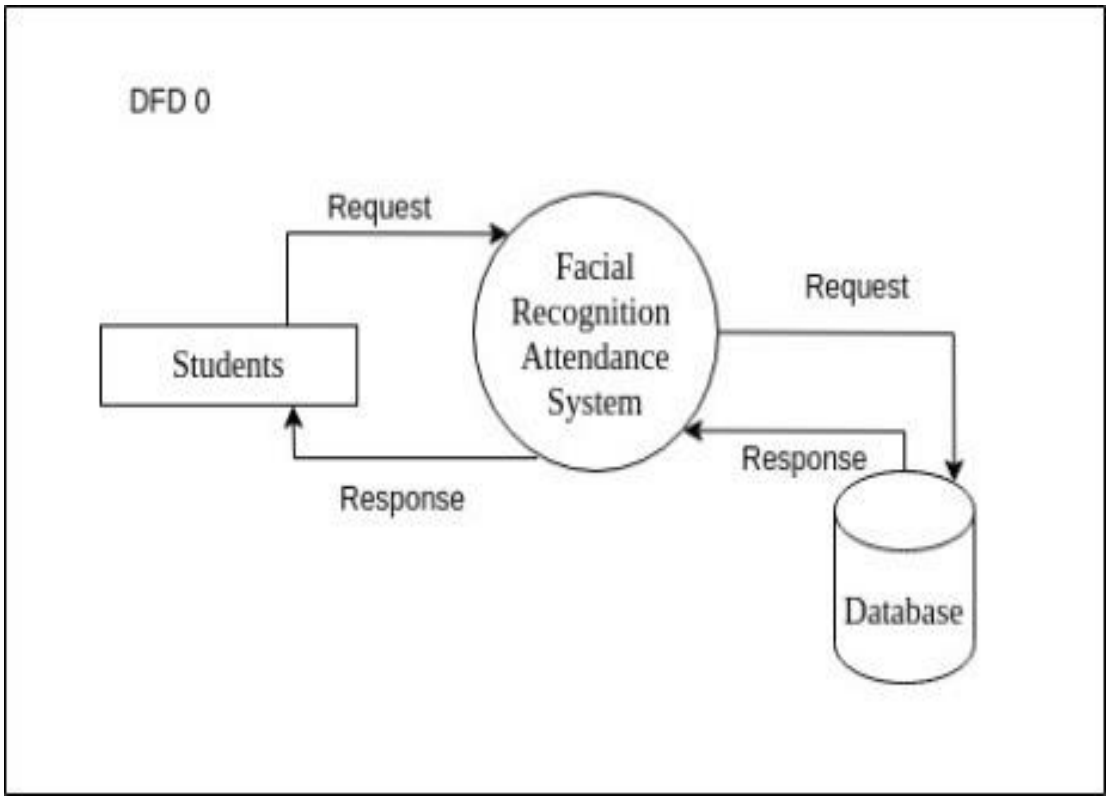


Figure-5.4.1 Level 0 Diagram

Level 1

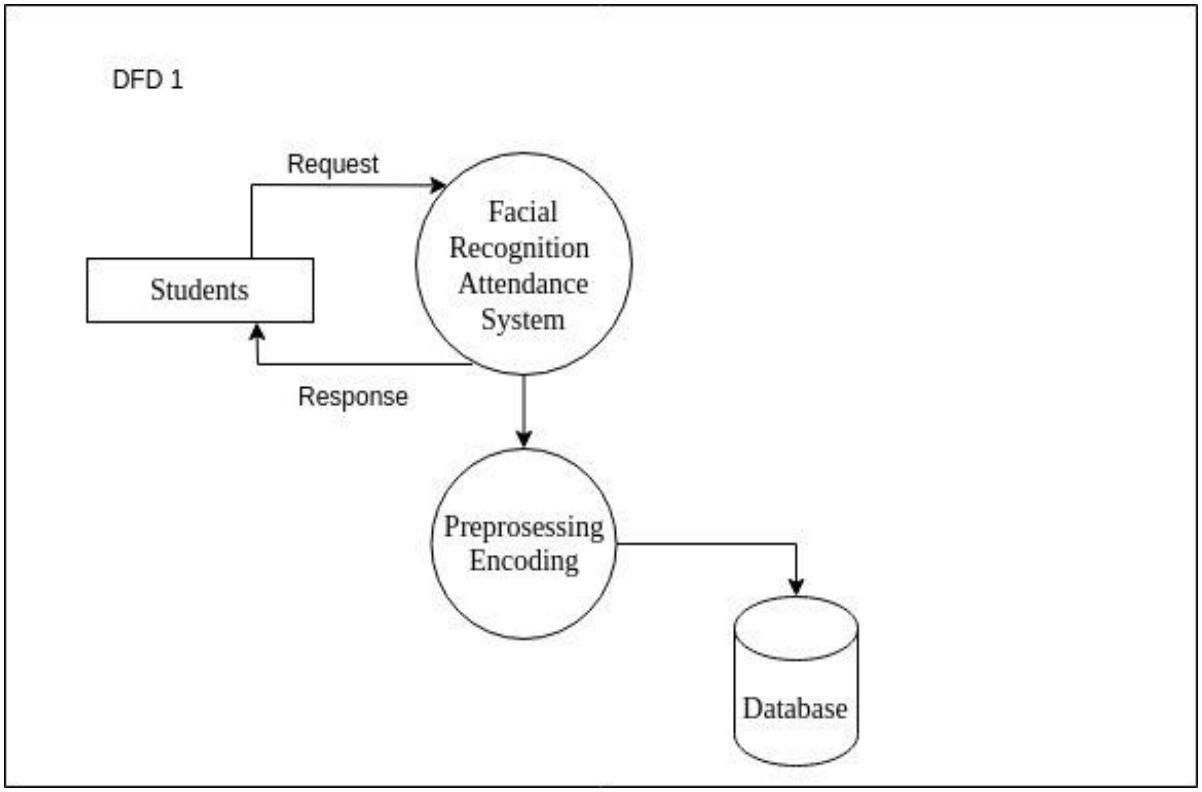


Figure-5.4.2 Level 1 Diagram

Level 2

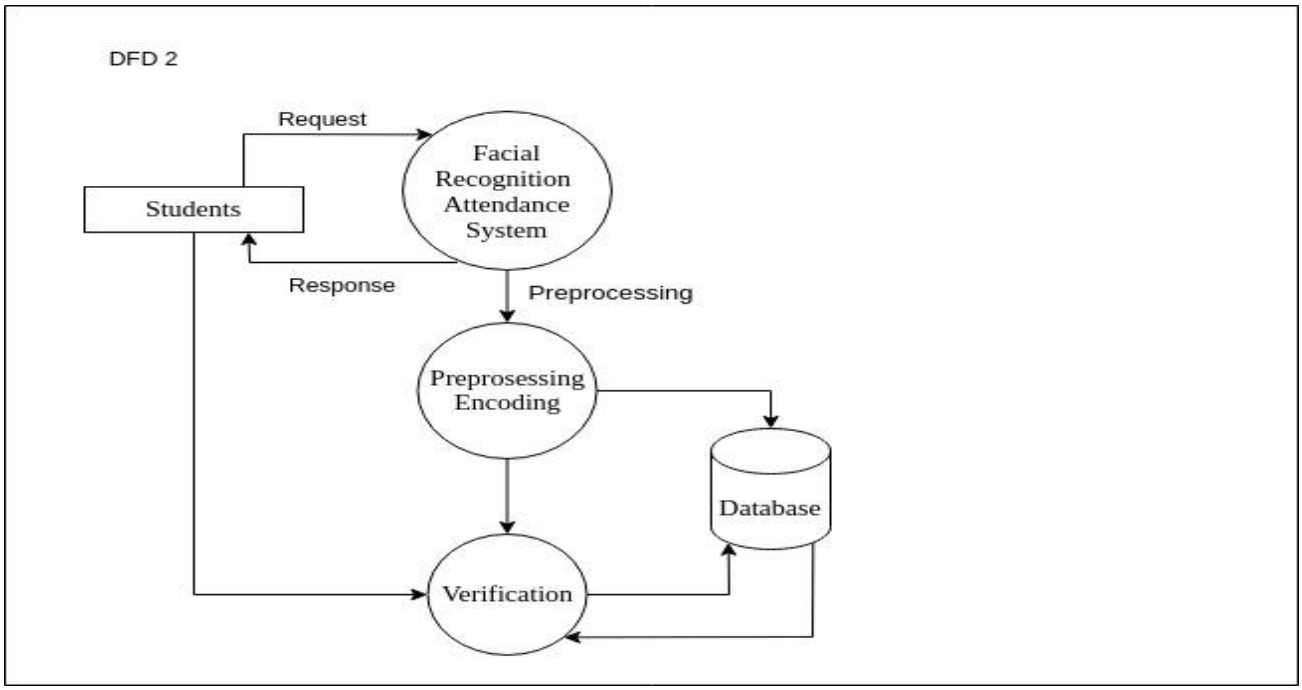


Figure-5.4.3 Level 2 Diagram

5.5**Advantages.**

1.**Accurate attendance tracking:** With face recognition technology, the attendance system can accurately track the attendance of students or employees without any manual intervention.

2.**Time-saving**: Face recognition attendance system saves time as it eliminates the need for manual attendance taking, thereby reducing administrative tasks.
3.**Cost-effective:** It is a cost-effective solution as it reduces the need for traditional attendance systems such as biometric attendance systems or swipe cards.

4.**Increased security:** With face recognition technology, the system can verify the identity of the person in real-time, thereby increasing security levels.

5.**Eliminates proxy attendance:** The face recognition attendance system eliminates the possibility of proxy attendance, as it can accurately identify and match the individual with their unique facial features.

6.**Easy integration:** The system can be easily integrated with other software and hardware systems, making it a flexible and versatile solution.

7.**Real-time monitoring:** The system can monitor attendance in real-time, allowing for timely interventions in case of any discrepancies or issues.

Overall, a face recognition attendance system using Python and OpenCV is a reliable, accurate, and efficient solution for attendance tracking in various settings, including schools, colleges, and organizations.

CHAPTER 6

Conclusion

5. Conclusion

Before the development of this project. There are many loopholes in the previous method while taking attendance using old method which caused many troubles to most of the institutions.

Therefore, the facial recognition feature method is secure enough, reliable and available for use. It saves time and lot of effort.

A face recognition attendance system using Python and OpenCV is an efficient and reliable solution for automating the attendance process. The system utilizes advanced image processing techniques to capture, detect, and recognize faces, thereby providing an accurate and efficient way to mark attendance.

The system can be implemented in various settings, such as educational institutions, workplaces, and public events, to streamline the attendance process and improve efficiency. It also eliminates the need for manual record-keeping, thus reducing the chances of errors and fraudulent attendance.

Moreover, the system can generate reports and provide real-time attendance data, enabling administrators to monitor attendance and track the progress of students or employees easily. The implementation of such a system can lead to significant improvements in the management of attendance, ultimately leading to better academic or organizational outcomes.

Overall, the facial recognition feature method is secure enough, reliable and available for use. It saves time and lot of effort.

References

Journal Papers:-

[1]. Yang, Hao, and Xiaofeng Han. "Face recognition attendance system based on real-time video processing." IEEE Access 8 (2020)
[2]. Suresh, V., S. Chakravarthi Dumpa, C. Deepak Vankayala, H. Aduri, and J. Rapa. "Facial recognition attendance system using python and OpenCv." Quest Journals-Journal of Software Engineering and Simulation 5, no. 2 (2019)
[3]. Ravindranadh, J., and Peram Venkata Ravindra Reddy. "PYTHON Based Attendance System." (2020)
[4]. Gupta, Naman, Purushottam Sharma, Vikas Deep, and Vinod Kumar Shukla. "Automated attendance system using OpenCV." In (2020) 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)
[5]. Solanki, Kamini, and Prashant Pittalia. "Review of face recognition techniques." International Journal of Computer Applications 133, no. 12 (2016): 20-24.

Useful Links:-

1.Article: Advanced Attendance System.
Website: **https://en.wikipedia.org/wiki/Facial_recognition_system**
2.Learning Python and Firebase Database video tutorials.
Website: **https://youtu.be/iBomaK2ARyI**
https://youtu.be/LaGYxQWYmmc
https://www.w3schools.com/

3. Advanced attendance System.
Website: **https://www.slideshare.net/**