**Chapter 1**

**INTRODUCTION**

* 1. **The history of face reorganization**

Facial recognition is more than 50 years old. A research team led by Woodrow W Bledsoe ran experiments between 1964 and 1966 to see whether 'programming computers' could recognize human faces. The American researchers Bledsoe et al. studied facial recognition computer programming. They imagine a semi-automatic method, where operators are asked to enter twenty computer measures, such as the size of the mouth or the eyes. 2005: The Face Recognition Grand Challenge competition was launched to encourage and develop face recognition technology designed to support existent facial recognition initiatives. 2011: Everything accelerates due to deep learning, a machine learning method based on artificial neural networks. The computer selects the points to be compared: it learns better when it supplies more images. 2014: Facebook knows how to recognize faces due to its internal algorithm. The social network claims that its method approaches the performance of the human eye near to 97% Today, facial recognition technology advancement has encouraged multiple investments in commercial, industrial, legal, and governmental applications. For example: In its new updates, Apple introduced a facial recognition application where its implementation has extended to retail and banking.Mastercard developed the Selfie Pay, a facial recognition framework for online transactions.From 2019, people in China who want to buy a new phone will now consent to have their faces checked by the operator.Chinesepolice used a smart monitoring system based on live facial recognition; using this system, they arrested, in 2018, a suspect of “economic crime” at a concert where his face, listed in a national database, was identified in a crowd of 50,000 persons.

* 1. **Face Reorganization**

A face analyser is software that identifies or confirms a person's identity using their face. It works by identifying and measuring facial features in an image. Facial recognition can identify human faces in images or videos, determine if the face in two images belongs to the same person, or search for a face among a large collection of existing images. Biometric security systems use facial recognition to uniquely identify individuals during user onboarding or logins as well as strengthen user authentication activity. Mobile and personal devices also commonly use face analyzer technology for device security. Face recognition systems use computer algorithms to pick out specific, distinctive details about a person’s face. These details, such as distance between the eyes or shape of the chin, are then converted into a mathematical representation and compared to data on other faces collected in a face recognition database. The data about a particular face is often called a face template and is distinct from a photograph because it’s designed to only include certain details that can be used to distinguish one face from another. Some face recognition systems, instead of positively identifying an unknown person, are designed to calculate a probability match score between the unknown person and specific face templates stored in the database. These systems will offer up several potential matches, ranked in order of likelihood of correct identification, instead of just returning a single result. Face recognition systems vary in their ability to identify people under challenging conditions such as poor lighting, low quality image resolution, and suboptimal angle of view.

Recognition algorithms can be divided into two main approaches

* **Geometric:** Is based on geometric relationship between facial landmarks, or in other words the spatial configuration of facial features. That means that the main geometric features of the face such as eyes, nose and mouth are first located and then faces are classified on the basis of various geometric distance and angles between features.
* **Photometric stereo:** Used to recover the shape of an object from a number of images taken under different lighting conditions. The shape of the recovered object is defined by a gradient map, which is made up of an array of surface normal.

**1.2.1 Different approaches of Face Reorganization**

Face recognition has attracted researches in different backgrounds such as face recognition, face pattern, neural networks, computer vision, computer graphics and psychology. It is challenging method, but yet it is interesting. Some of the face recognition methods are

1. Holistic matching
2. Feature-based matching
3. Hybrid matching

* **Holistic matching methods:** In this method entire face is used as a raw input to a recognition system. The holistic matching method can be classified into linear and non-linear projection methods. Linear projection appearance-based method includes principal component analysis, independent component analysis, linear discriminate analysis and linear regression classifier. Nonlinear projection appearance- based method includes kernel principal component analysis, kernel linear discriminate analysis, and locally linear embedding. In nonlinear approach the input image is mapped into higher dimensional space in which the face is simplified and linear. Hence traditional methods are applied. In principal component analysis a number of images are taken using grey levels. Each image is mapped to a long vector of grey levels. Several views of each person are collected in the database during training. During recognition a vector corresponding to an unknown face is compared with all vectors in the database. The vector image in the database which is closest to the unknown face is declared as a recognised face. The dimensionality of each vector will be very large. The grey levels are also too sensitive to noise and lighting conditions. A possible solution to these problems are reducing the dimensionality of a space by finding principal component to space the face and only a few significant eigen vectors can be used to represent a face thus reducing the dimensionality. In the first case a training set of size nxn is created. After creating training set, each image is converted into vectors
* **Feature based matching method:** In this method the eyes, nose, mouth is located and features are extracted which are feed to the structural classifier. In feature-based method face restoration is a big challenge. Due to large variations the system is unable to retrieve features. The extraction methods can be distinguished into General methods based on eyes, ears, and nose Feature-template based method, Structural matching methods. In digital image processing and computer vision local binary pattern histogram approach is used to recognise a feature of a face. This approach was first demonstrated in the year 1990. In texture classification LBP method is found very influential. The detection performance can be regularly and precisely be improved by combining the LBP and Histogram descriptor. Each pixel of an image is assigned an operator. For a 3\*3 pixel, a threshold value is set in which all the neighbouring values are compared to the central value and if the value is greater than the central value than it is represented as one, else zero. Local regions are formed and texture descriptor is extracted from these regions. These features are concatenated to farm global description of the face. Spatially enhanced histogram is formed from these global patterns. To measure the distance spatial enhanced histogram is used. To measure the distance weighted chi square is used. To create an LBP feature vector the window is divided into cells and each pixel value in the cell is compared with neighbouring pixel value. If the pixel value of the central cell is less than its neighbouring cells value than it encoded as zero. If the pixel value of the central cell is greater than its neighbouring cells value than it encoded as one.
* **Hybrid matching method:** It is a combination of holistic method and feature based matching method. In this method 3-D images are used. This allows the system to note the curves of eyes, nose, cheeks etc. full face can be constructed as the depth and axis of measurement gives a lot of information about the face. This method involves detection of the face either by scanning or by photograph taken in real time. The location, angle and size of the head is positioned. Each curve is measured and a template is made and the region outside the eye, inside the eye, region of the nose is focused. This template is than converted into code. This code is stored in the database and then later compared with input image for recognition.
  1. **Face Detection**

In face analysis, face detection uses facial expressions to identify which parts of an image or video should be focused on to determine age, gender and emotions. In a facial recognition system, face detection data is required to generate a faceprint and match it with other stored faceprints. face detection algorithms typically start by searching for human eyes, one of the easiest features to detect. They then try to detect facial landmarks, such as eyebrows, mouth, nose, nostrils and irises. Once the algorithm concludes that it has found a facial region, it does additional tests to confirm that it has detected a face.

* 1. **Motivation and Theoretical overview**

In the recent years, Image processing which deals with extracting useful information from a digital image plays a unique role in the advent of technological advancements. It focusses on two tasks

* Improvement of pictorial information for human interpretation
* Processing of image data for storage, transmission and representation for autonomous machine perception. Also, people have started to use image capturing devices never as before with the advent of smart phones and closed-circuit television. Since the application of image processing is vast, extensive work and research have been carrying out in utilizing its potential to and to make new innovative applications.
* Facial recognition has been the earliest of the application derived from this technology, which is one of the most fool proof methods in human detection. Face is a typical multidimensional structure and needs good computational analysis for recognition.
* Biometrics methods have been used for the same purpose since a long time now. Although it is effective, it is still not completely reliable for purpose of detecting a person.

**1.5 Problem Statement**

Attendances of every student are being maintained by every school, college and university Empirical evidences have shown that there is a significant correlation between students’ attendances and their academic performances. There was also a claim stated that the students who have poor attendance records will generally link to poor retention. Therefore, faculty has to maintain proper record for the attendance. The manual attendance record system is not efficient and requires more time to arrange record and to calculate the average attendance of each student. Hence there is a requirement of a system that will solve the problem of student record arrangement and student average attendance calculation. One alternative to make student attendance system automatic is provided by facial recognition.

**1.6 Aims and Objectives**

The objective of this project is to develop face recognition attendance system. Expected achievements in order to fulfil the objectives are:

* To detect the face segment from the video frame.
* To extract the useful features from the face detected.
* To classify the features in order to recognize the face detected.
* To record the attendance of the identified student.
* Detection of unique face image amidst the other natural components such as walls, backgrounds etc.
* Extraction of unique characteristic features of a face useful for face recognition.
* Detection of faces amongst other face characters such as beard, spectacles etc.
* Effective recognition of unique faces in a crowd (individual recognition in crowd).
* Automated update in the database without human intervention.

**Chapter 2**

**LITERATURE SURVEY**

Anushka Waingankar et al. (2018) Face Recognition based Attendance Management System using Machine Learning In this paper we proposed an automated attendance management system which tackles the predicament of recognition of faces in biometric systems subject to different real time scenarios such as illumination, rotation and scaling. This model incorporates a camera that captures input image, an algorithm to detect a face from the input image, encode it and recognize the face and mark the attendance in a spreadsheet and convert it into PDF file. [1]

Dr. V Suresh et al. (2019) Facial Recognition Attendance System Using Python and OpenCVthe main purpose of this project is to build a face recognition-based attendance monitoring system for educational institution to enhance and upgrade the current attendance system into more efficient and effective as compared to before. [2]

Fuzail et al. (2019) Face Detection System for Attendance of Class Students have proposed automated student attendance system based on face recognition. The proposed systems were used a camera to capture student faces and mark the attendance. [3]

Clyde Gomes et al. (2020) Class Attendance Management System using Facial Recognition proposed However, not all cameras can be used to capture the image of palm vein. Furthermore, the accuracy of palm vein recognition in attendance system proposed in was only78%. Therefore, from the current state of the art of automated attendance system it can be found that face recognition is the best approach to recognize student in an attendance system. [4]

Harshad Patil et al. (2021) Automatic Attendance System Using Face Recognition the paper proposed a web-based student attendance system that uses face recognition. In the proposed system, Convolutional Neural Network (CNN) is used to detect faces in images, deep metric learning is used to produce facial embedding, and to classify student's faces. Thus, the computer can recognize faces. [5]

**Chapter 3**

**SYSTEM ANALYSIS**

* 1. **Existing System**

The attendance Management can be done in several ways. the belove listed technologies are existing systems of attendance management system every technology has a unique feature the detailed information of existing system is provided in this chapter.

* + 1. **Signature Based System**

The objective of signature verification is to identify the unique characteristics of person’s writing style. It has been widespread nowadays because the collection of the signature is non-invasive and people are also familiar with the signature in their daily life. Handwritten Signature is different from the biometric system is that to identify the correct signature of a particular person. In this system, the user provides signature samples and then classify that signature that it is a genuine or forgery. Forgery Signature can be classified into three types: Random, Skilled and Simple forgeries. In Random forgery, the forger has no information about user signature and randomly done the signature it will present an Overall shape of signature. In simple forgery, forgery knows the user but has no idea about user’s signature and do sign in his/her style. In Skilled forgery, forgery knows about user’s signature information so to identify skilled forgeries are very hard to identify. Skilled forgery made more practices of user’s signature to do forgery signature of the user. The signature can be acquired in two forms: Online and Offline. In Online signature is acquired by digital devices like it can be biometric technology that signature can be done on digitized devices like tablet or phone. The user can do signature by the digital pen in tablet etc. In Offline, the user needs to do signature in their attendance sheet and then each signature is scanned and sampled in the individual signature.



**Fig 3.1: Signature Based Attendance system**

* + 1. **Fingerprint based System**

Fingerprint identification is one of the most crucial building blocks for smart interactions. Amongst the identification methods, fingerprint recognition is identified to be the most natural ones, that uses to identify people in day by day lives. Although other methods, such as magnetic cards, can provide enhanced performance, those are not appropriate for natural smart interactions due to their intrusive nature. In comparison, fingerprint recognition provides passive identification that is the person to be identified does not need to cooperate or take any specific action. Attendance record plays an important role in the academic achievement of institute students. Attendances of all students are being maintained by every school, college and university.

Attendance Management Falls into two categories Namely:

* Conventional
* Automated Methods.

The manual attendance record system is not efficient and requires more time to arrange record and to calculate the average attendance of each student. Hence there is a requisite of a system that will solve the problem of student record arrangement and student average attendance calculation. Faculty has to maintain proper record for the attendance. As a result, fingerprint recognition is used to mark the attendance of the employees as well as student. This system provides flexibility optimizing the attendance of the students and the employees at the same time separately. The Fingerprint authentication has many advantages such as very high accuracy, the most economical biometric PC user authentication technique.



**Fig 3.2: Fingerprint Based attendance management system**

* + 1. **Iris Recognition**

Iris recognition verification is one of the most reliable personal identification methods in biometrics. In the beginning, the idea of using iris patterns for personal identification was originally proposed in 1936 by ophthalmologist Frank Burch. By the 1980's the idea had appeared in James Bond films, but it still remained science fiction and conjecture. As biometric of human for identification purpose which cannot be stolen or lost. From the biometric system there exist different types of biometric such as thumb recognition, palm recognition, face recognition and iris recognition etc. Amongst which the iris is more preferred. The reason for the popularity of iris recognition verifying is the uniqueness, stability, permanency and easily taking. Iris recognition system is highly protected and stable that results in a single enrolment for the lifetime. The unique pattern on the surface of the iris is formed during the first year of life. Formation of the unique patterns of the iris is random and not related to any genetic factors. The only characteristic that is dependent on genetics is the pigmentation of the iris, which determines its color. Due to the epigenetic nature of iris patterns, the two eyes of an individual contain completely independent iris patterns, and identical twins possess uncorrelated iris patterns.

the following basic modules:

* Image acquisition, iris location, and pre-processing,
* Iris texture feature extraction and signature encoding,
* Iris signature matching for recognition or verification.



**Fig 3.3: Iris Based attendance management system**

* + 1. **RFID based System**

Radio-frequency identification (RFID) is a technology that uses radio waves to transfer data from an electronic tag, called RFID tag or label, connected to an object, via a reader for the cause of identifying and monitoring the object. Radio frequency identification (RFID) is a matured technological know-how that accommodates he use of electromagnetic or electrostatic coupling in the radio frequency portion of the electromagnetic spectrum to uniquely become aware of an object, animal, or person. RFID chips include a radio transmitter that emits a coded identification number when queried through a reader device. Some RFID tags can be examined from various meters away and beyond the line of sight of the reader. The utility of bulk reading allows an almost-parallel analyzing of tags. This small type is incorporated in client products, and even implanted in pets, for identification. The tag's facts are stored electronically. The RFID tag includes a small RF transmitter which transmits an encoded radio sign to interrogate the tag, and receiver which receives the message and responds with its identification information. Some RFID tags do not use a battery. Instead, the tag makes use of the radio power transmitted by way of the reader as its strength source. The RFID machine plan consists of an approach of discriminating countless tags that might be within the range of the RFID reader. RFID can be used in many applications. A tag can be affixed to any object and used to and manage inventory, assets, people, etc. For example, it can be affixed to cars, laptop equipment, books, cellular phones, etc. The RFID attendance device is an automated embedded system used in taking attendance of registered individuals in a specific organization.



**Fig 3.4: RFID Based attendance management system**

**3.2 Proposed System**

****

**Fig 3.5: Face Recognition Based attendance management system**

**3.2.1 Image Enhancement**

Color images often contain background clutter that reduces the accuracy of face detection and facial recognition systems. To improve this, we have methods to remove that unneeded color data from a color input image before recognition performs on the clear grayscale version. The result was efficient, fast, and accurate processing of millions of facial images for the application. Image cropping removes unnecessary surrounding material from the images for some specific reason. Image post-processing can help to extract relevant data. For example, many extraction methods are used in face detection systems to ensure the face in the image crop is in the most suitable position.

**3.2.2 Face Detection**

Face detection uses machine learning ([ML](https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML)) and artificial neural network ([ANN](https://www.techtarget.com/searchenterpriseai/definition/neural-network)) technology, and plays an important role in face tracking, face analysis and [facial recognition](https://www.techtarget.com/searchenterpriseai/definition/facial-recognition). In face analysis, face detection uses facial expressions to identify which parts of an image or video should be focused on to determine age, gender and emotions. In a facial recognition system, face detection data is required to generate a faceprint and match it with other stored faceprints.

**3.2.3 Feature Extraction**

Feature extraction is the application of extracting algorithm on digital images to reduce redundancy and irrelevancy present in the image. The main goals of feature extraction are to reduce the time of machine training and complexity of space, in order to achieve a dimension reduction. Feature extraction algorithms transform an input data into the set of features, meanwhile select features containing the most relevant information from the original data. Feature extraction maintain acceptable classification accuracy by reducing maximum number of irrelevant features.

**3.2.4 Face Database**

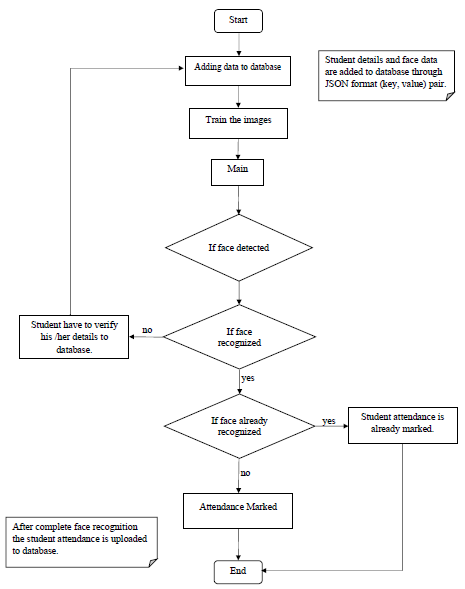
The captured or detected faces are stored in the real-time face database. If a person trying to store his face in the database, then 50 images of face angels are stored. At the time of recognition, a person’s face is compared with those 50 images to identify that face is matched successfully.

**3.2.5 Face Recognition**

Face recognition is a method of identifying or verifying the identity of an individual using their face. Face recognition systems can be used to identify people in photos, video, or in real-time.

**Chapter 4**

**SYSTEM DESIGN**

****

**Fig 4.1: System Design of Face Recognition attendance system**

In the above flow diagram, first block describes the process of adding student data to the database. Means all the student related data (Name, USN, Branch, Present year) with student images are added to database in JSON format in this case we are using firebase database. In second block images are trained with eigen face algorithm and face details are encoded. The face Recognition process start with main block, if the camera detects faces then recognize the faces if the face is known then system check whether the student face already recognized or not. A message box appears informing that the student attendance is marked.

**Chapter 5**

**REQUIREMENT SPECIFICATION**

## 5.1 Hardware Requirements

The hardware required for the development of this project is:

|  |  |
| --- | --- |
| Processor | : Intel i3 or higher version |
| RAM | : Minimum 8GB |
| ROM | : Minimum 128 SSD |
| GPU | : AMD Radeon or Other graphics |
| Devices | : Monitor, Keyboard, Mouse, Camera |

## 5.2 Software Requirements

The software required for the development of this project is:

|  |  |
| --- | --- |
| Software | : PyCharm community version 2023.1.1 |
| Operating System | : Windows 10 or 11 |
| Programming Language | : Python 3.7 version |
| Libraries | : OpenCV, dlib, numpy, etc |
| Database | : Real-time Database |
| Server | : Firebase |

**Chapter 6**

**IMPLEMENTATION**

**6.1 Project Implementation**

This system deals with the maintenance of the student's attendance details. It generates the attendance of the student on basis of presence in class. It is maintained on the daily basis of their attendance.

**6.1.1 Libraries**

* **OpenCV**

In our proposed system we will detect and recognize the faces by using Eigen object detector algorithm. This can be done with the help of OpenCV with haar cascades which are present in the OpenCV inbuilt. We will design an attendance system for the students with the help of this OpenCV, for this the system will need a HD webcam to take the input images in a fixed area where the camera is located. The images which are taken from the camera are detected with haarcascade frontal faces and eyes then trained with Eigen algorithm, the trained faces are stored in a database first and compared to the trained images (the trained images are initially present in the database means the related persons to the particular college or organization) after comparing it will make attendance to the recognized persons. A. Microsoft Visual C# the proposed system is implementing with this visual studio to build this system as an application. Visual Studio is a fully set of development tools to build ASP.NET Web applications, XML Web Services, desktop applications, and mobile applications. Visual Basic, Visual C#, and Visual C++ all use the same integrated development environment (IDE).

OpenCV [open-source Computer vision] The OpenCV is an open-source computer vision library. This library was written in C and C++ and runs under different operating systems like windows, Mac OS and Mac OS. The main goal of the opencv is to provide an easy way of computer vision infrastructure which helps to build the applications easily based on vision One of OpenCV’s goals is to provide a simple-to-use computer vision infrastructure that helps people build fairly sophisticated vision applications quickly. Here we are using different libraries of opencv the used libraries are as fallows They are different algorithms are there to do this kind of tasks, but here we are using the Eigen face recognizer. This is better than some of the emerging algorithms. pen source computer vision library (OpenCV) One of OpenCV’s goals is to provide a simple-to-use computer vision infrastructure that helps people build fairly sophisticated vision applications quickly. OpenCV library contains over 500 functions that span many areas in vision. The primary technology behind Face recognition is OpenCV; the interface is designed using FLTK. The user stands in front of the camera keeping a minimum distance of 50cm and his image is taken as an input. The frontal face is extracted from the image then converted to gray scale and stored. The principal component Analysis (PCA) algorithm is performed on the images band the eigen values are stored in an xml file. When a user requests for recognition the frontal face is extracted from the captured video frame through the camera. The eigen value is re-calculated for the test face and it is matched with the stored data for the closest neighbor

**OpenCV's application areas include:**

* 2D and 3D feature toolkits
* Ego motion estimation
* Facial recognition system
* Gesture recognition
* Human–computer interaction (HCI)
* Mobile robotics
* Motion understanding
* Object identification
* Segmentation and recognition
* Stereopsis stereo vision: depth perception from 2 cameras
* Structure from motion (SFM)
* Motion tracking
* Augmented reality
* **dlib**

Dlib is a general purpose [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) software [library](https://en.wikipedia.org/wiki/Library_(computing)) written in the programming language [C++](https://en.wikipedia.org/wiki/C%2B%2B).Its design is heavily influenced by ideas from [design by contract](https://en.wikipedia.org/wiki/Design_by_contract) and [component-based software engineering](https://en.wikipedia.org/wiki/Component-based_software_engineering). Thus, it is, first and foremost, a set of independent software components. It is [open-source software](https://en.wikipedia.org/wiki/Open-source_software) released under a [Boost Software License](https://en.wikipedia.org/wiki/Boost_(C%2B%2B_libraries)#License). Since development began in 2002, Dlib has grown to include a wide variety of tools. As of 2016, it contains software components for dealing with [networking](https://en.wikipedia.org/wiki/Computer_network), [threads](https://en.wikipedia.org/wiki/Thread_(computing)), [graphical user interfaces](https://en.wikipedia.org/wiki/Graphical_user_interface), [data structures](https://en.wikipedia.org/wiki/Data_structure), [linear algebra](https://en.wikipedia.org/wiki/Linear_algebra), [machine learning](https://en.wikipedia.org/wiki/Machine_learning), [image processing](https://en.wikipedia.org/wiki/Image_processing), [data mining](https://en.wikipedia.org/wiki/Data_mining), [XML](https://en.wikipedia.org/wiki/XML) and text parsing, [numerical optimization](https://en.wikipedia.org/wiki/Numerical_optimization), [Bayesian networks](https://en.wikipedia.org/wiki/Bayesian_network), and many other tasks. In recent years, much of the development has been focused on creating a broad set of statistical machine learning tools and in 2009 Dlib was published in the [Journal of Machine Learning Research](https://en.wikipedia.org/wiki/Journal_of_Machine_Learning_Research).[[2]](https://en.wikipedia.org/wiki/Dlib#cite_note-2) Since then it has been used in a wide range of domains.

* **NumPy**

NumPy is a package that defines a multi-dimensional array object and associated fast math functions that operate on it. It also provides simple routines for linear algebra and fft and sophisticated random-number generation. NumPy replaces both Numeric and Numarray.

**Example demonstrating NumPy:** from numpy import \* from PIL import Image ar = ones ((100,100), float32) ar = ar \* 100 for i in range (0,100): ar[i,:] = 100 + (i \* 1.5) im = Image.fromarray(ar,"F")

The numpy namespace includes all names under the numpy.core and numpy.lib namespaces as well. Thus, import numpy will also import the names from numpy.core and numpy.lib. This is the recommended way to use numpy.

**6.2 Code Implementation**

The project is mainly developed by using the python programming language. In this phase the code which is used to develop the project is described.

**6.2.1 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

cred = credentials.Certificate("serviceAccountKey.json")

firebase\_admin.initialize\_app(cred, {

'databaseURL': "",

'storageBucket': ""

})

bucket = storage.bucket()

cap = cv2.VideoCapture(1)

cap.set(3, 640)

cap.set(4, 480)

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

# Importing the mode images into a list

folderModePath = 'Resources/Modes'

modePathList = os.listdir(folderModePath)

imgModeList = []

for path in modePathList:

imgModeList.append(cv2.imread(os.path.join(folderModePath, path)))

# print(len(imgModeList))

# Load the encoding file

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

imgStudent = []

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:

# Get the Data

studentInfo = db.reference(f'Students/{id}').get()

print(studentInfo)

# Get the Image from the storage

blob = bucket.get\_blob(f'Images/{id}.png')

array = np.frombuffer(blob.download\_as\_string(), np.uint8)

imgStudent = cv2.imdecode(array, cv2.COLOR\_BGRA2BGR)

# Update data of attendance

datetimeObject = datetime.strptime(studentInfo['last\_attendance\_time'],

"%Y-%m-%d %H:%M:%S")

secondsElapsed = (datetime.now() - datetimeObject).total\_seconds()

print(secondsElapsed)

if secondsElapsed > 30:

ref = db.reference(f'Students/{id}')

studentInfo['total\_attendance'] += 1

ref.child('total\_attendance').set(studentInfo['total\_attendance'])

ref.child('last\_attendance\_time').set(datetime.now().strftime("%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']), (1025, 625),

cv2.FONT\_HERSHEY\_COMPLEX, 0.6, (100, 100, 100), 1)

cv2.putText(imgBackground, str(studentInfo['starting\_year']), (1125, 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] = imgStudent

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("Webcam", img)

cv2.imshow("Face Attendance", imgBackground)

cv2.waitKey(1)

**6.2.2 EncodeGenerator.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': "",

'storageBucket': ""

})

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])

fileName = f'{folderPath}/{path}'

bucket = storage.bucket()

blob = bucket.blob(fileName)

blob.upload\_from\_filename(fileName)

print(studentIds)

def findEncodings(imagesList):

encodeList = []

for img in imagesList:

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

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

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()

print("File Saved")

**6.2.3 AddDatatoDatabase.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': ""

})

ref = db.reference('Students')

data = {

"321654":

{

"name": "Murtaza Hassan",

"major": "Robotics",

"starting\_year": 2017,

"total\_attendance": 7,

"standing": "G",

"year": 4,

"last\_attendance\_time": "2022-12-11 00:54:34"

},

"852741":

{

"name": "Emly Blunt",

"major": "Economics",

"starting\_year": 2021,

"total\_attendance": 12,

"standing": "B",

"year": 1,

"last\_attendance\_time": "2022-12-11 00:54:34"

},

"963852":

{

"name": "Elon Musk",

"major": "Physics",

"starting\_year": 2020,

"total\_attendance": 7,

"standing": "G",

"year": 2,

"last\_attendance\_time": "2022-12-11 00:54:34"

}

}

for key, value in data.items():

ref.child(key).set(value)

**Chapter 7**

**TESTING**

**7.1 Unit Testing**

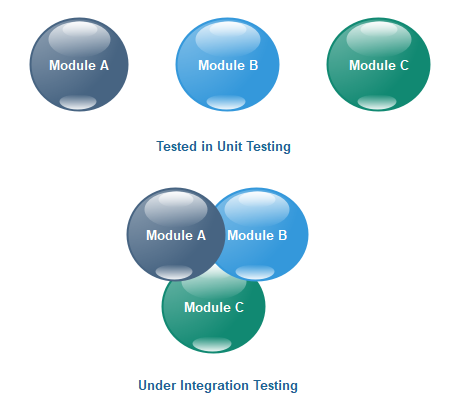
In computer programming, unit testing is a software testing method by which individual units of source code is tested. The goal of unit testing is to validate individual components. As come to the face recognition attendance management system testing there are 3 different individual units which are tested in this unit testing phase the

**Fig 7.1: Unit Testing of Face Recognition attendance system**

In the above figure the first unit add data to database is tested first in which we are going to add the details of students with their face data. At the time of testing this unit if the details of students are not added to the database that leads to the failure of unit testing. Otherwise, the details are added successfully which means the first unit testing is completed after that it will move to the next unit testing which is encoder generator and another one face recognizer is also tested if all the units are working properly then testing will move to next phase.

**7.2 Integration Testing**

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements.



**Fig 7.2: Integration Testing of Face Recognition attendance system**

In the above integration testing phase, the units which are tested in the unit testing phase those are integrated here with each other and tested for the errors if any error found then it leads to a failure of testing if not then it is successful.

**7.3 System Testing**

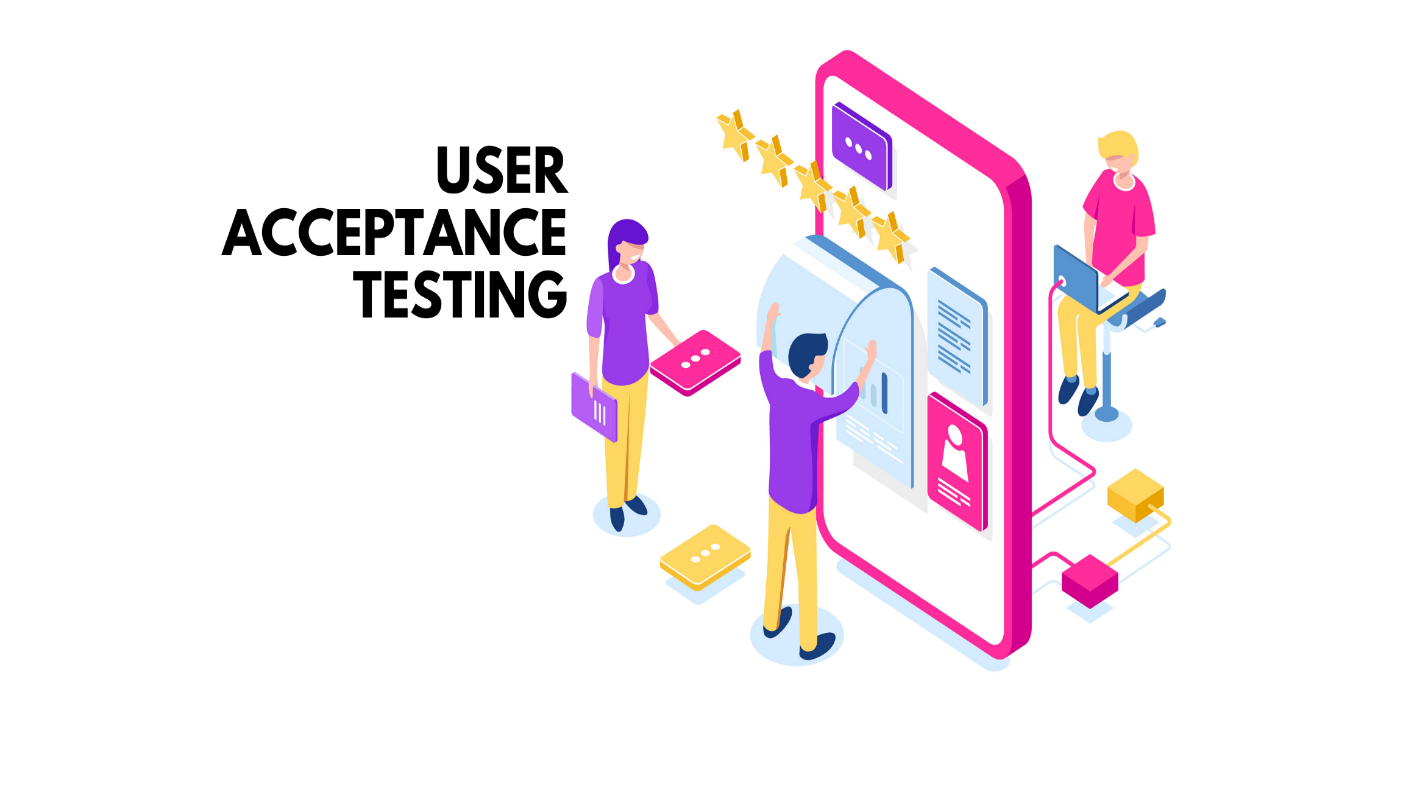
In the system testing phase, the software undergoes its first test as a complete, integrated application to determine how well it carries out its purpose. For this, the developers pass the software to independent testers who were not involved in its development to ensure that the testing results from impartial evaluations. System testing is vital because it ensures that the software meets the requirements as determined by the client.



**Fig 7.3: System Testing of Face Recognition attendance system**

**7.4 Acceptance Testing**

Acceptance Testing is the last phase of software testing performed after System Testing and before making the system available for actual use. It is a formal testing according to user needs, requirements and business processes conducted to determine whether a system satisfies the acceptance criteria or not and to enable the users, customers or other authorized entities to determine whether to accept the system or not.



**Fig 7.4: Acceptance Testing of Face Recognition attendance system**

**Chapter 8**

**ADVANTAGES**

* Ease in maintaining attendance.
* Reduced paper work.
* Automatically operated and accurate.
* Reliable and user friendly.
* **Automated attendance tracking:** Face recognition technology can automate attendance tracking, eliminating the need for manual processes like taking attendance sheets or using time clocks. This can save time and reduce errors in attendance records.
* **Accurate and reliable:** Face recognition technology can accurately and reliably identify individuals, even in low-light conditions, and with facial masks, as some systems can now detect the face behind masks. This can help reduce errors in attendance records and prevent time fraud.
* **Real-time monitoring:** Face recognition technology can provide real-time monitoring of attendance, which can help supervisors to identify attendance issues and take corrective measures as needed.
* **Improved security:** Face recognition technology can enhance the security of attendance records by ensuring that only authorized individuals are able to clock in or out. This can prevent time fraud and improve the accuracy of attendance records.
* **Cost-effective:** Face recognition technology can be a cost-effective solution for attendance management, as it eliminates the need for manual processes, such as paper-based attendance sheets, and reduces the time and resources required to manage attendance records.

**DISADVANTAGES**

* Privacy concerns: Face recognition technology raises concerns about privacy and data security. Some individuals may feel uncomfortable with their faces being recorded and stored, especially if they are not aware of the purpose of the technology or how their data is being used.
* Technical issues: Face recognition technology may encounter technical issues, such as difficulty in detecting faces in low-light conditions or if someone is wearing a mask. This can lead to inaccurate attendance records and frustration for employees.
* Training and maintenance: Face recognition technology requires specialized training and ongoing maintenance to ensure optimal performance. This can add to the overall cost and complexity of implementing the technology.

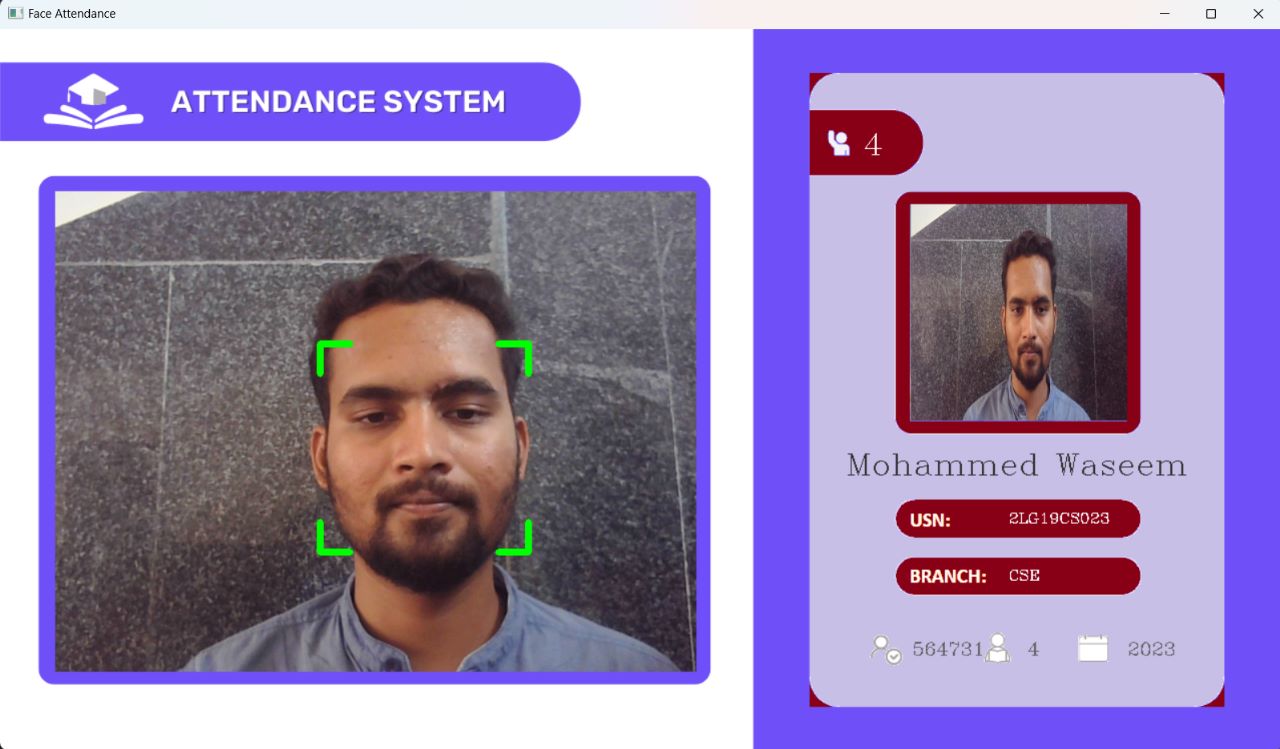
**Chapter 9**

**APPLICATONS**

* Workplace attendance management: Face recognition technology can be used to track employee attendance in workplaces, such as factories, offices, and retail stores. This can eliminate the need for manual processes and improve the accuracy of attendance records.
* School attendance management: Face recognition technology can be used in schools and universities to track student attendance. This can help educators to identify attendance issues and take corrective measures as needed.
* Event attendance management: Face recognition technology can be used to track attendance at events, such as conferences, concerts, and sports games. This can help event organizers to monitor attendance in real-time and ensure that attendees are authorized to enter the event.
* Healthcare attendance management: Face recognition technology can be used in healthcare settings, such as hospitals and clinics, to track patient attendance and reduce wait times. This can improve patient satisfaction and help healthcare providers to optimize their resources.
* Government attendance management: Face recognition technology can be used in government settings, such as voting booths or government offices, to track attendance and improve security. This can help prevent fraudulent attendance and improve accountability.

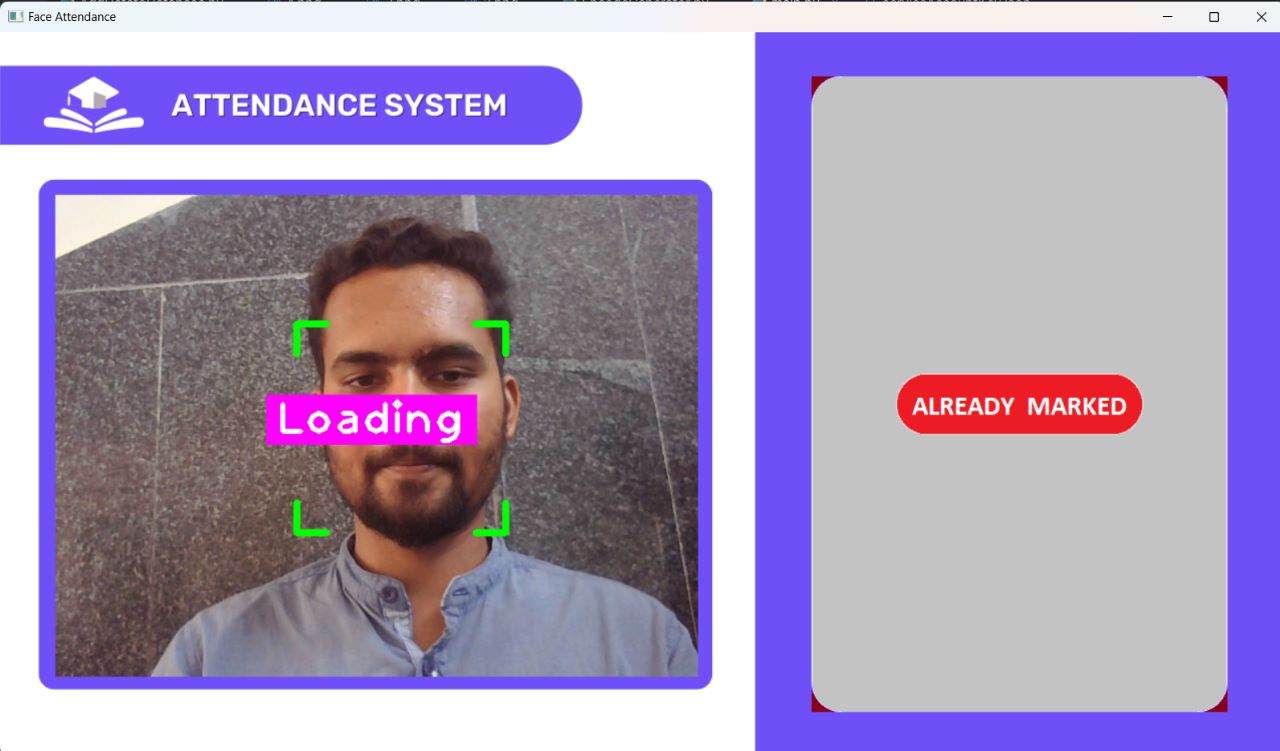
**Chapter 10**

**RESULTS**

****

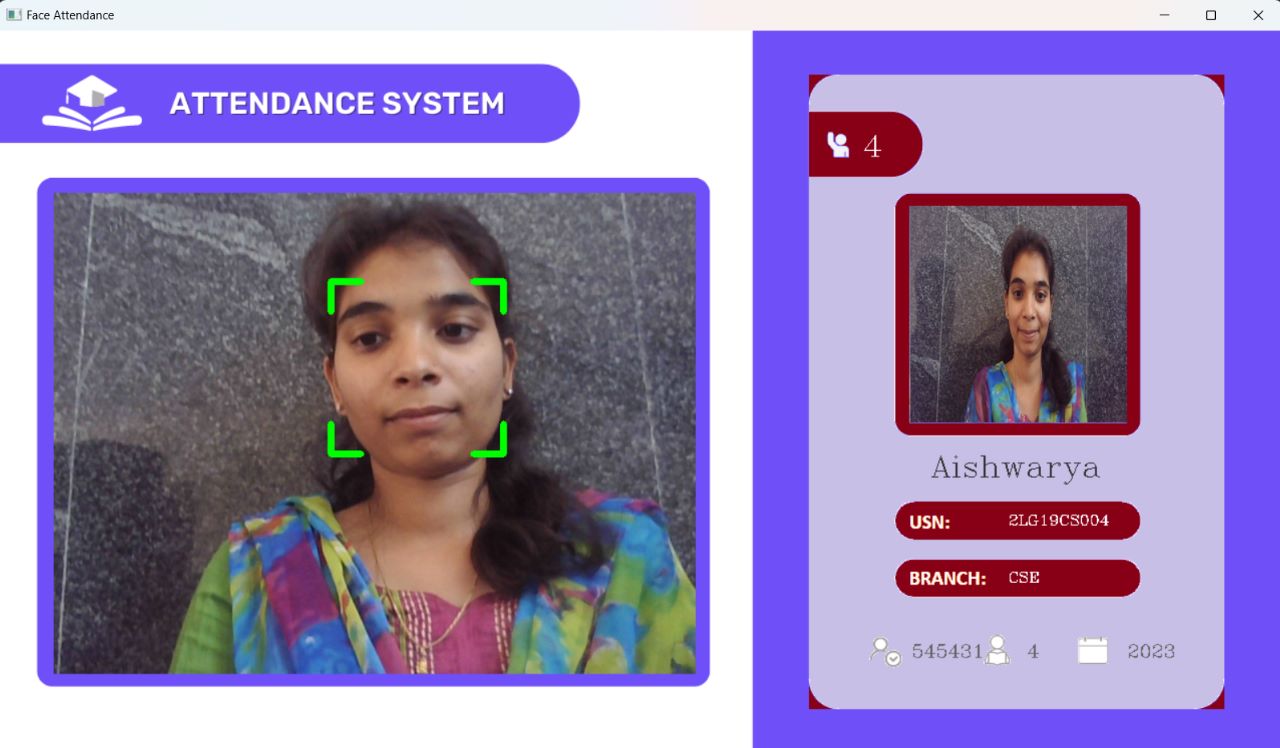
**Fig 10.1: Face Recognition student 1.**

The above snapshot shows the face Recognition of student 1 and his details like marked attendance, name, USN, branch, year.

****

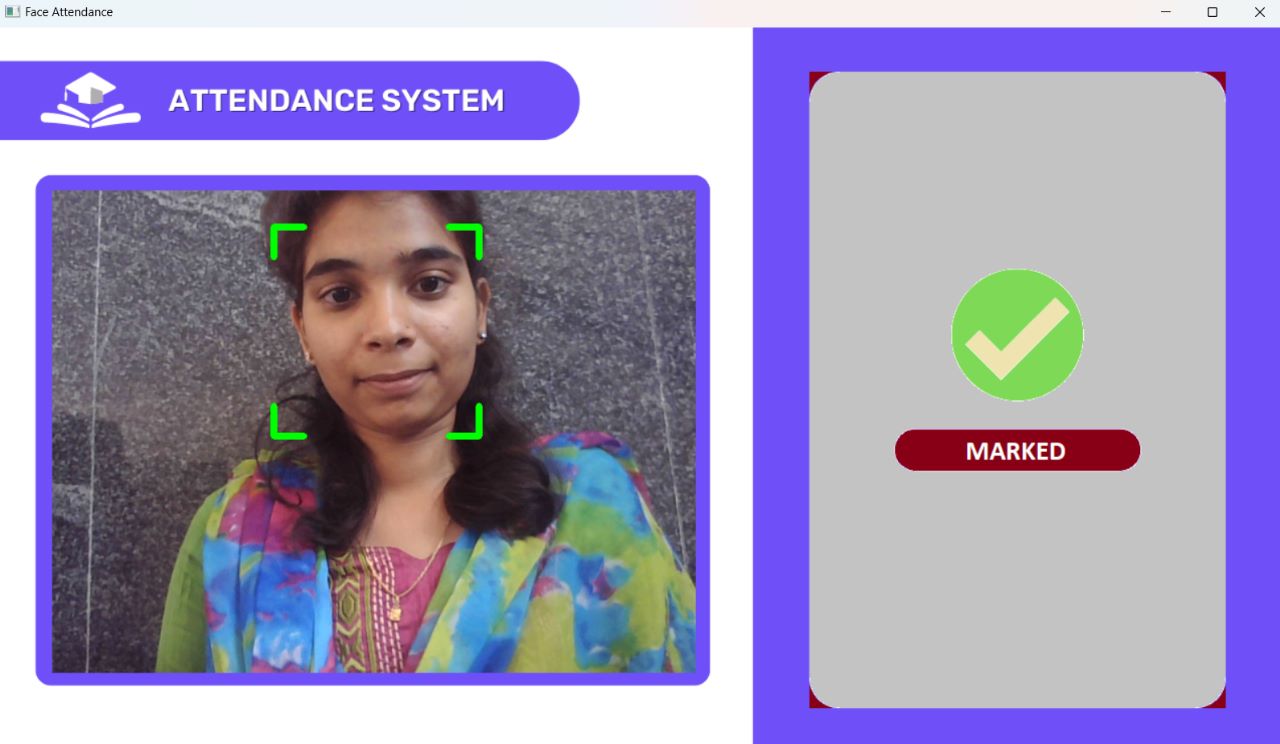
**Fig 10.2: Already Marked Interface.**

If a student trying to take attendance again than it shows that the attendance is already marked so that student cannot take more than one attendance in a day. A student cannot take more than one attendance in a specified time period.



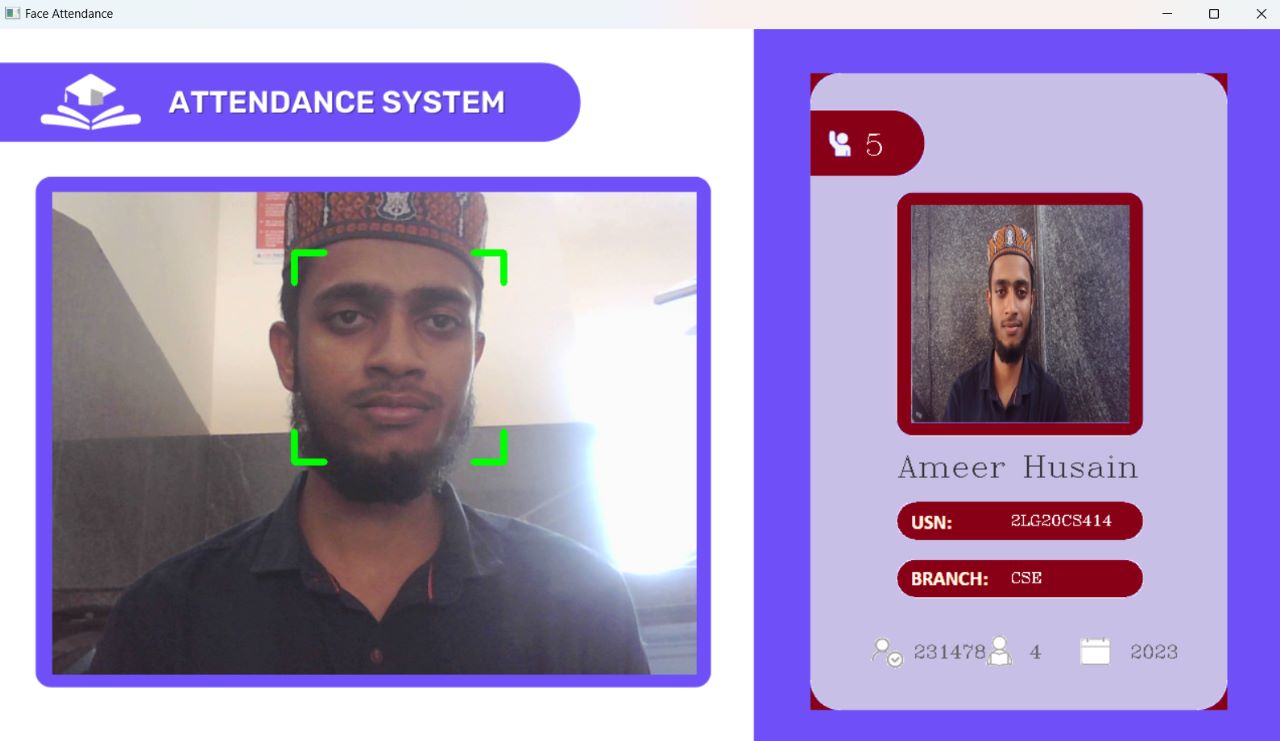
**Fig 10.3: Face Recognition student 2.**

The above snapshot shows the recognized face of student and his details like marked attendance, name, USN, branch, year.



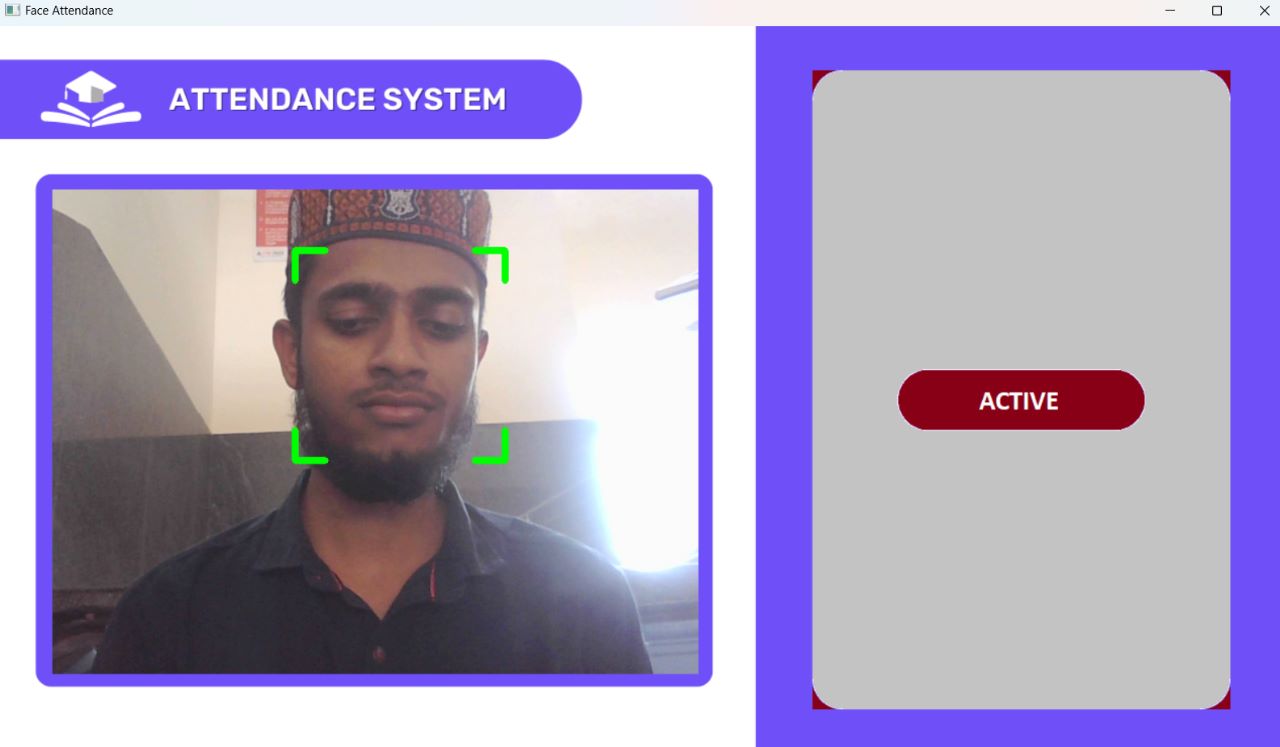
**Fig 10.4: Student Attendance Marked to Database.**

When student face is detected and face matches if the student face is present in database. Then the student attendance is marked for that specified day. And a message is appeared informing that student marked His/her attendance completely.

****

**Fig 10.5: Face Recognition student 3.**

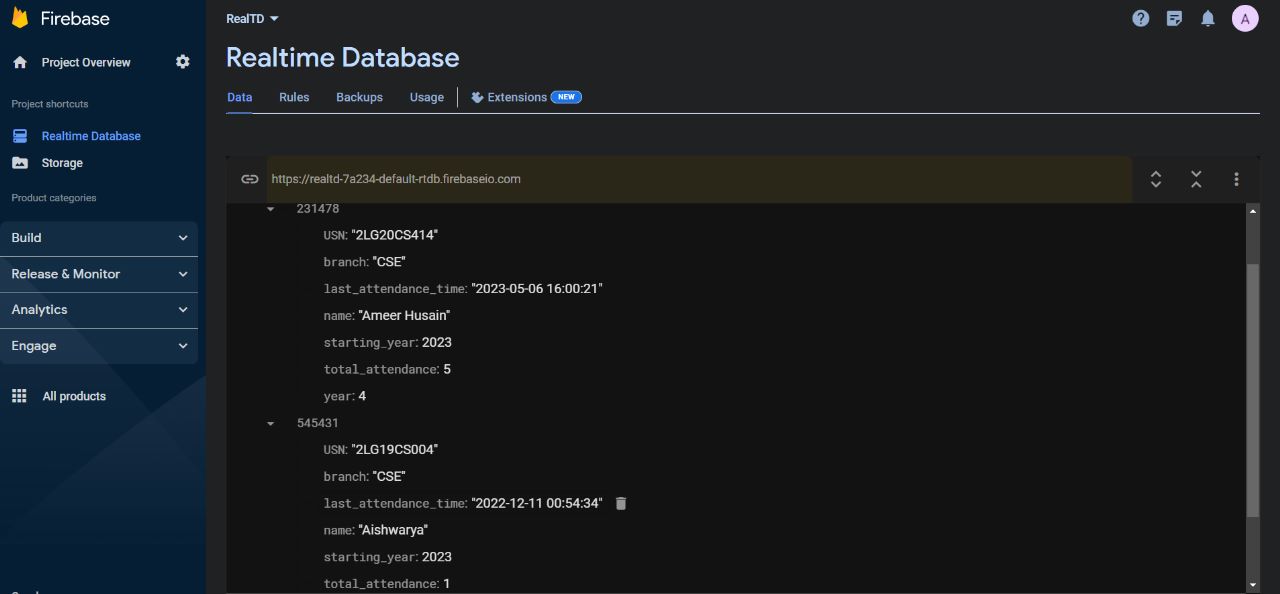
The above snapshot shows the recognized face of student and his details like marked attendance, name, USN, branch, year. These data are uploaded to database in json format to firebase.

****

**Fig 10.6: Face Activeness Interface.**

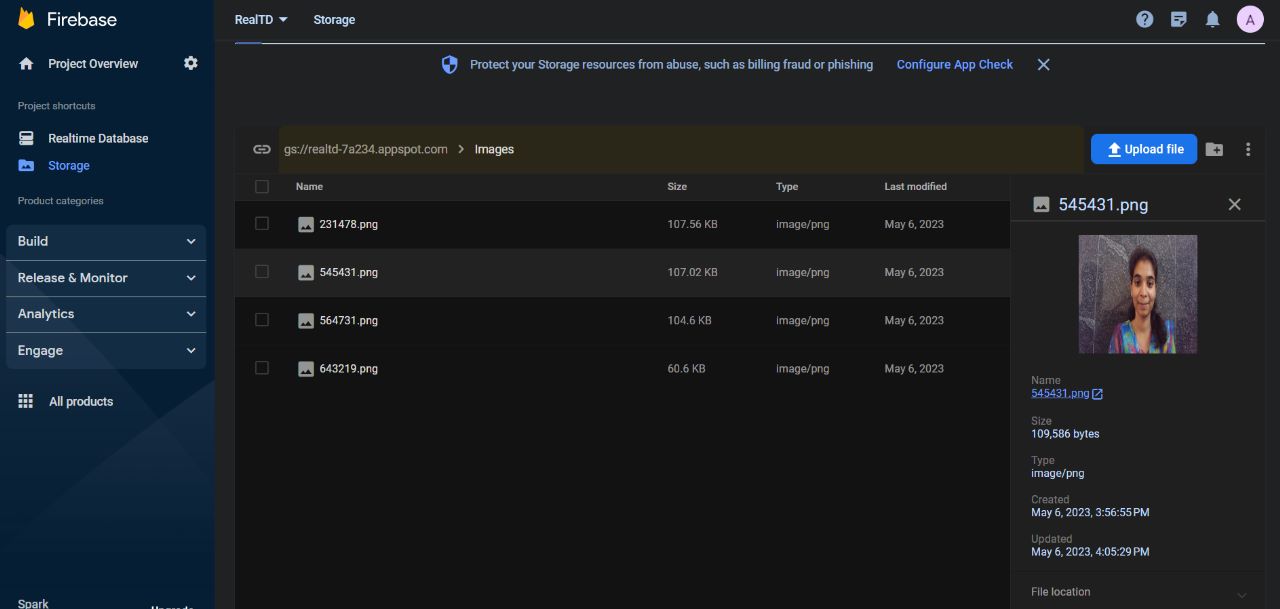
The Above Snapshot shows the Activeness of face which means if the student face detectable or face appears clearer there should be no more noises in live video that help a face Recognition process easier.

.



**Fig 10.7: Realtime Database Interface.**

The above snapshot shows the real time database that stores the student’s details like USN, Branch, Name, starting year, total attendance etc. it stores live instance of data when a student appears to give his attendance.



**Fig 10.8: Static Student’s Data Storage.**

The Above snapshot shows the Static metadata storage.it stores student images and data like image format (PNG, JPG), image size (216x216), date of creation and last updated information and also includes file location.