INTRODUCTION

The project is "Facial Recognition Attendance System". It will overcome the problem of old attendance-taking systems, where the teacher has to shout the name/Roll number of the students and mark it in a register. This process is long and time taking.
There are many ways to solve this problem, the most famous solution is the "Fingerprint Attendance System". In this system we have to save the fingerprint of every student, if the number of students is too large it becomes almost impossible because it is very tough to collect the fingerprint of every student. That's why this is not used in any institution.
"Facial Recognition Attendance System", As can be understood from the name, it will recognize faces through the camera and create an excel file that will store the attendance. User has to save images of the students with their name, which is convenient because every institution collects the images of the students for the ID cards. After saving images to the system user has to enter a subject for which the user wants to take attendance and after that camera will start and the system start taking attendance after completing the attendance it will store the attendance in an excel file and save it on a local device which is your computer.
In this project we use **Python** programming language with Supervised learning as a base.

Project Synopsis

- Statement about the problem:

According to the previous attendance management system, **the accuracy of the data** collected is the biggest issue. This is because the attendance might not be recorded personally by the original person, in another word, the attendance of a particular person can be taken by a third party without the realization of the institution which violates the accuracy of the data.

Normally in school and college teacher/lecturers uses traditional attendance system where teacher/lecturer shouts the role number of the students and write it down in a register which is very time taking.

There are many ways to solve this problem, the most famous solution is "Fingerprint Attendance System". In this we have to save fingerprint of every student, if the number of Student is too large it become almost impossible because it is very tough to collect the fingerprint of every student. That's why this is not used in any institution. Assuming the time taken for a student to sign his/her attendance on a name list is approximately 1 minute. In 1 hour, only approximately 60 students can sign their **attendance which is obviously inefficient and time consuming**.

- Why is the particular topic chosen?

Well our topic is chosen because of the fact that attendance is very important for any institution such that the process is traditionally weak. So for the upgradation of the process we have made an python based project in regards to the fact that no external involvement effects the attendance data and the data is stored in a per day manner excel sheet to be checked whenever needed.

- **Methodologies:**

The methodology of this project is mainly comparison of stored data and the raw data so as the raw data can be identified by the system.The Old attendance taking system is time taking and exhausting process and its alternative option "Fingerprint Attendance System" is also that much of a nuisance where we have to save fingerprint of every student, which sometimes become impossible if the number of students is too large.

In “Facial Recognition Attendance System” we save images of every student as an stored data, which is more convenient than “Fingerprint Attendance System” because every institution collects the images for the ID cards. This system tries to recognize the person who are standing in front of the camera and forms a raw data, if his image is already present in stored data it will recognize him4 and store it, if it is not available then it will show “Unknown”. This process is done by the Machine Learning, more specifically we only use Supervised learning.
In simple term it is simply comparing the image to its database which is a folder of images and show the result.

• What contribution would the project make?

Well as we have discussed the problem related to the traditional attendance system so this project will make a pathway or an upgradation to this system.

This project will show us a new way to modernize our old system by providing an interface between a student and a computer.

By this project we can show the younger generation about a small but creative idea about how we can change a process of Attendance to an computer application by programming.

Theoretical Background

According to Literature, Student Attendance System by Face Detection, maintaining attendance is very important and compulsory in all the institutes for checking the performance of students. Every institute has its method in this regard. Some are taking attendance manually using the old paper old file-based approach and some have adopted methods of automatic attendance using some biometric techniques.

An automated attendance system based on face recognition is a biometric system where typically, it registers the attendance of each student present in a class by detecting and identifying all of their faces, and then this recorded information is ideally transmitted to a server device which may compute the attendance of each student and store and update the corresponding data in a database. Automated attendance systems are more reliable, rigid, and efficient than the traditional attendance systems and other biometric attendance systems, leading to better productivity and output of both the teachers and students, as well as better consumption of time.

An Automatic Attendance System Using Image Processing, maintaining attendance is very important and compulsory in all the institutes for checking the performance of students. Every institute has its method in this regard.
Some are taking attendance manually using the old paper or file-based approach and some have adopted methods of automatic attendance using some biometric techniques. There are many automatic methods available for this purpose i.e. biometric attendance. All these methods also waste time because students have to make a queue to touch their thumb on the scanning device.

Face detection and recognition section detect face from the image capture by the camera, and the image of the face is crop and store. The element recognizes the images of student’s faces, which have been registered manually with their names and ID code in the record. Face recognition data and face identification data are verification into the record. Automatic face recognition (AFR) technology has seen a remarkable improvement in presentation over the past years, and such systems are now widely used for safety and marketable applications. An automatic system for human face recognition in a real-time environment for a university to mark the attendance of its
employees.

Objective of the project

The main objective of the 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 recognising 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 and is directly mailed to the respected faculty.

- Functionalities Provided by Facial recognition attendance system are as follows:-

▪ The targeted groups of the attendance monitoring system are the students and staff of an educational institution.

▪ The facial recognition process can only be done for 1 person at a time.

▪ An excel sheet is created which contains the student attendance and is mailed to the respected faculty.

▪ The project has to work under a Wi-Fi coverage area or under Ethernet connection, as the system need to update the database of the attendance system constantly.

▪ The device on which the application is running is powered up by power bank to improve the portability of the application.

Scope of the project

- The facial recognition process can be done for 1 person and also for a group of persons if the camera quality is good.
- The project is offline it takes attendance and record it to a Excel sheet (.csv file).
- The project's main motto is for the time conservation for attendance process and also for no third party interference with attendance data.

About Python

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems. This versatility, along with its beginner-friendliness, has made it one of the most-used programming languages today. A survey conducted by industry analyst firm RedMonk found that it was the second-most popular programming language among developers in 2021.

History of Python :

Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle-detecting garbage collection, reference counting, and Unicode support. Python 3.0, released in 2008, was a major revision that is not completely backward-compatible with earlier versions. Python 2 was discontinued with version 2.7.18 in 2020.

Why is Python needed?

- Web development (server-side),
- Software development,
- Mathematics
- System scripting.

Uses of Python :

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Advantages:

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-oriented way or a functional way.

Python Syntax:

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Example

```
print("Hello, World!")
```

Libraries used:

Tkinter:

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

- Import the *Tkinter* module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

OpenCV(cv2):

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e whatever operations one can do in Numpy can be combined with OpenCV. Arsenal i.e whatever operations one can do in Numpy can be combined with OpenCV.

Face Recognition:

Recognize and manipulate faces from Python or from the command line with the world's simplest face recognition library. Built using dlib's state-of-the-art face recognition built with deep learning. The model has an accuracy of 99.38% on the Labeled Faces in the Wild benchmark. This also provides a simple face_recognition command line tool that lets you do face recognition on a folder of images from the command line.

Numpy:

NumPy is a Python library used for working with arrays.

It also has functions for working in domain of linear algebra, fourier transform, and matrices.

NumPy was created in 2005 by Travis Oliphant. It is an opensource project and you can use it freely.

NumPy stands for Numerical Python.

NumPy is a Python library and is written partially in Python, but most of the parts that require fast computation are written in C or C++.

In Python we have lists that serve the purpose of arrays, but they are slow to process.

NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.

The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.

Arrays are very frequently used in data science, where speed and resources are very important.

OS:

Python OS module provides the facility to establish the interaction between the user and the operating system. It offers many useful OS functions that are used to perform OS-based tasks and get related information about operating system.

The OS comes under Python's standard utility modules. This module offers a portable way of using operating system dependent functionality.

The Python OS module lets us work with the files and directories.

Glob:

In Python, we have many in-built modules for performing various tasks, and one of such tasks we want to perform with the Python modules is finding and locating all the files present in our system, which follows a similar pattern. This similar pattern can be a file extension, the file name's prefix, or any similarity between two or many files. We have many different Python modules with which we can easily perform this task using a Python program, but not all the modules are as efficient as others. In this tutorial, we are going to learn about one of such efficient modules, i.e., glob module in Python, with which we can perform file matching with a specific pattern by using it inside a program. We will learn in detail about the glob module in Python, how we can use it inside a program, what its key features are and the application of this module.

With the help of the Python glob module, we can search for all the path names which are looking for files matching a specific pattern (which is defined by us). The specified pattern for file matching is defined according to the rules dictated by the Unix shell. The result obtained by following these rules for a specific pattern file matching is returned in the arbitrary order in the output of the program. While using the file matching pattern, we have to fulfil some requirements of the glob module because the module can travel through the list of the files at some location in our local disk. The module will mostly go through those lists of the files in the disk that follow a specific pattern only.

Following are some listed applications of the Python glob module, and we can use this module in the given functions:

1. Sometimes, we want to search for a file that has a certain prefix in its name, any common string in the middle of the names of many files or have the same certain extension. Now, to perform this task, we may have to write a code that will scan the whole directory and then it will produce the result. Instead of it, the glob module is going to be very helpful in this case as we can use the functions of the glob module and perform this task very easily and can save our time.

2. Other than this, the Glob module is also very useful when one of our programs have to look for the list of all the files in a given file system with the names of the files matching a similar pattern. Glob module can easily perform this task and that too without opening the result of the program in other sub-shell.

DateTime:

In Python, date and time are not a data type of their own, but a module named datetime can be imported to work with the date as well as time. Python Datetime module comes built into Python, so there is no need to install it externally.

Python Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.

The DateTime module is categorized into 6 main classes –

- date – An idealized naive date, assuming the current Gregorian calendar always was, and always will be, in effect. Its attributes are year, month and day.

- time – An idealized time, independent of any particular day, assuming that every day has exactly 24*60*60 seconds. Its attributes are hour, minute, second, microsecond, and tzinfo

.

- datetime – Its a combination of date and time along with the attributes year, month, day, hour, minute, second, microsecond, and tzinfo.

- timedelta – A duration expressing the difference between two date, time, or datetime instances to microsecond resolution.

- tzinfo – It provides time zone information objects.

- timezone – A class that implements the tzinfo abstract base class as a fixed offset from the UTC (New in version 3.2).

Shutil:

Python shutil module provides the facility to perform the high-level file operation. It can operate with the file object and offers us the ability of copy and remove the files. It handles the low-level semantic such creating and closing file objects after performing all operations.

Python shutil module comes with the many built-in methods. We will explore a few important methods. To start working with this module, first we need to import it in our current Python file.

Shutil module in Python provides many functions of high-level operations on files and collections of files. It comes under Python's standard utility modules. This module helps in automating process of copying and removal of files and directories.

shutil.copy() method in Python is used to copy the content of source file to destination file or directory. It also preserves the file's permission mode but other metadata of the file like the file's creation and modification times is not preserved.

Source must represent a file but destination can be a file or a directory. If the destination is a directory then the file will be copied into destination using the base filename from source. Also, destination must be writable. If destination is a file and already exists then it will be replaced with the source file otherwise a new file will be created.

Imghdr:

The imghdr module in Python's standard library determines the type of image contained in a file or byte stream. There is only one function defined in imghdr module

imghdr.what(filename, h=None):

This function tests the image data contained in the file and returns a string describing the image type. The function also accepts h parameter. If given the filename is ignored and h is treated as the byte stream to test.

The imghdr module recognizes the following image types

value	image format
'rgb'	SGI ImgLib Files
'gif'	GIF 87a and 89a Files
'pbm'	Portable Bitmap Files
pgm'	Portable Graymap Files
'ppm'	Portable Pixmap Files
'tiff'	TIFF Files
'rast'	Sun Raster Files
'xbm'	X Bitmap Files
'jpeg'	JPEG data in JFIF or Exif formats
'bmp'	BMP files
'png'	Portable Network Graphics
'webp'	WebP files
'exr'	OpenEXR Files

Re:

This module provides regular expression matching operations similar to those found in Perl.

Both patterns and strings to be searched can be Unicode strings (str) as well as 8-bit strings (bytes). However, Unicode strings and 8-bit strings cannot be mixed: that is, you cannot match a Unicode string with a byte pattern or viceversa; similarly, when asking for a substitution, the replacement string must be of the same type as both the pattern and the search string.

Regular expressions use the backslash character ('\') to indicate special forms or to allow special characters to be used without invoking their special meaning. This collides with Python’s usage of the same character for the same purpose in string literals; for example, to match a literal backslash, one might have to write '\\\\' as the pattern string, because the regular expression must be \\, and each backslash must be expressed as \\ inside a regular Python string literal. Also, please note that any invalid escape sequences in Python’s usage of the backslash in string literals now generate a DeprecationWarning and in the future this will become a SyntaxError. This behaviour will happen even if it is a valid escape sequence for a regular expression.

The solution is to use Python’s raw string notation for regular expression patterns; backslashes are not handled in any special way in a string literal prefixed with 'r'. So r"\n" is a two-character string containing '\' and 'n', while "\n" is a one-character string containing a newline. Usually patterns will be expressed in Python code using this raw string notation.

It is important to note that most regular expression operations are available as module-level functions and methods on compiled regular expressions. The functions are shortcuts that don’t require you to compile a regex object first, but miss some fine-tuning parameters.

About Supervised Learning

Supervised learning is the types of machine learning in which machines are trained using well "labelled" training data, and on basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.

In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher.

Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to **find a mapping function to map the input variable(x) with the output variable(y)**.

In the real-world, supervised learning can be used for **Risk Assessment, Image classification, Fraud Detection, spam filtering**, etc.

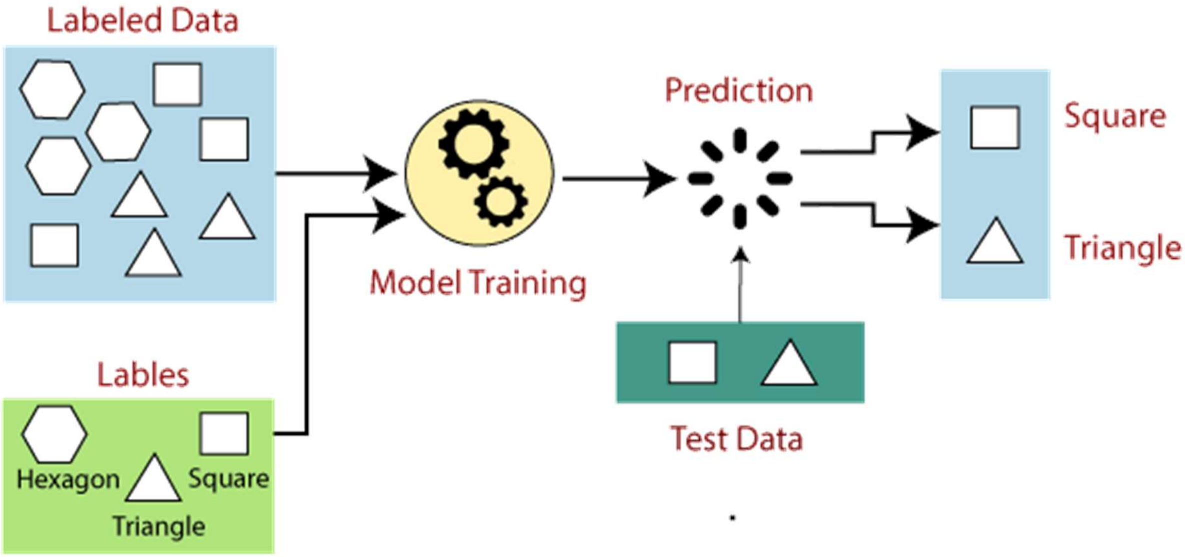
• Steps Involved in Supervised Learning:

- First Determine the type of training dataset
- Collect/Gather the labelled training data.
- Split the training dataset into training **dataset, test dataset, and validation dataset**.
- Determine the input features of the training dataset, which should have enough knowledge so that the model can accurately predict the output.
- Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc.
- Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.
- Evaluate the accuracy of the model by providing the test set. If the model predicts the correct output, which means our model is accurate.

• How Supervised Learning Works?

In supervised learning, models are trained using labelled dataset, where the model learns about each type of data. Once the training process is completed, the model is tested on the basis of test data (a subset of the training set), and then it predicts the output.

The working of Supervised learning can be easily understood by the below example and diagram:



Suppose we have a dataset of different types of shapes which includes square, rectangle, triangle, and Polygon. Now the first step is that we need to train the model for each shape.

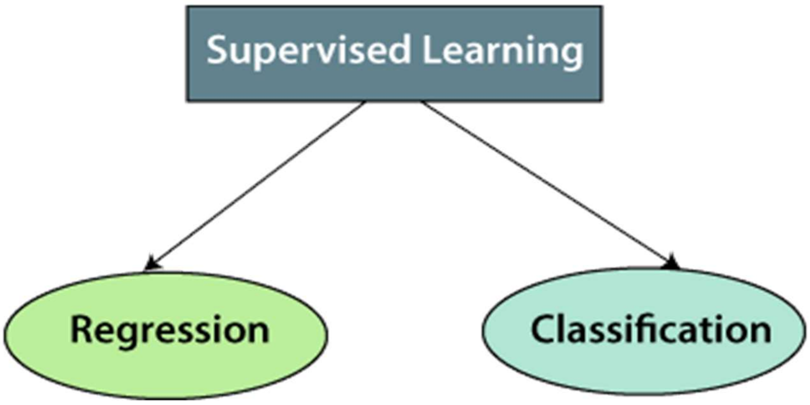
- If the given shape has four sides, and all the sides are equal, then it will be labelled as a **Square**.
- If the given shape has three sides, then it will be labelled as a **triangle**.
- If the given shape has six equal sides then it will be labelled as **hexagon**.

Now, after training, we test our model using the test set, and the task of the model is to identify the shape.

The machine is already trained on all types of shapes, and when it finds a new shape, it classifies the shape on the bases of a number of sides, and predicts the output.

- <u>**Types of supervised Machine learning Algorithms:**</u>

Supervised learning can be further divided into two types of problems:



1. Regression

Regression algorithms are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc. Below are some popular Regression algorithms which come under supervised learning:

- Linear Regression
- Regression Trees
- Non-Linear Regression
- Bayesian Linear Regression
- Polynomial Regression

2. Classification

Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, etc.

- Random Forest
- Decision Trees
- Logistic Regression
- Support vector Machines

Advantages of Supervised learning:

- With the help of supervised learning, the model can predict the output on the basis of prior experiences.
- In supervised learning, we can have an exact idea about the classes of objects.
- Supervised learning model helps us to solve various real-world problems such as **fraud detection, spam filtering**, etc.

Disadvantages of supervised learning:

- Supervised learning models are not suitable for handling the complex tasks.
- Supervised learning cannot predict the correct output if the test data is different from the training dataset.
- Training required lots of computation times.
- In supervised learning, we need enough knowledge about the classes of object.

Specifications

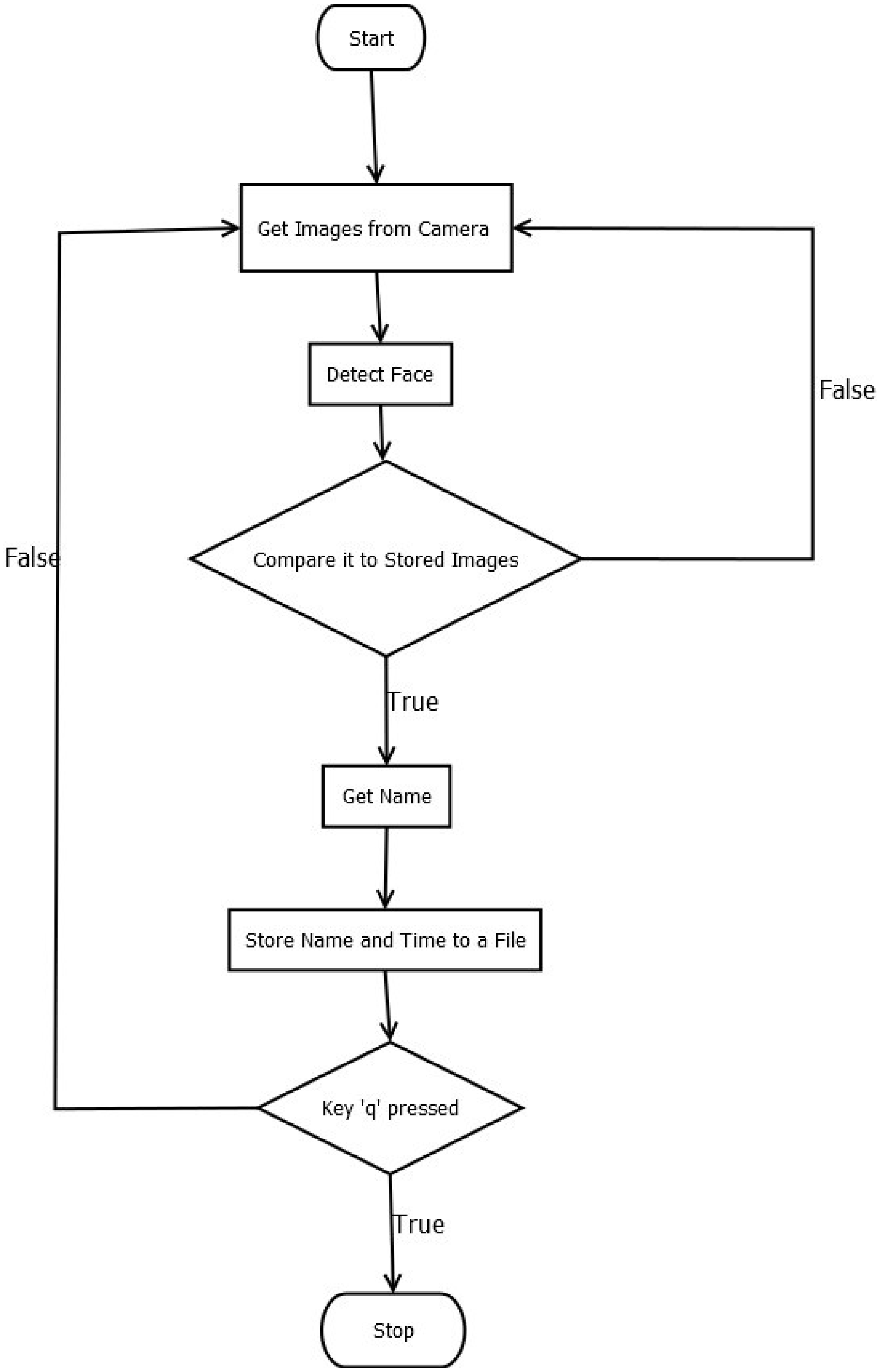
Hardware :

- Processor: Intel® i3
- RAM: 2GB
- Hard Disk: 250MB
- System Type:32-bit operating system
- Best Quality camera

Software:

- Language: Python 3.9.6
- Software Development kit: Visual Studio code

Flow-Chart Diagram



Methodology Adopted

❖ HOG Algorithm(Histogram Oriented Gradients):

The algorithm for extracting HOGs begins by counting occurrences of gradient orientation in localized portions of an image. Basically, the image is divided into small connected regions, called cells, and for each cell compiling a histogram of gradient directions or edge orientations for the pixels within the cell. The histogram counts are normalized so as to compensate for illumination. The combination of these histograms then represents the descriptor. Invariance to scale and rotation is also achieved by extracting descriptors only from salient points (keypoints) in the scale space of the image. The steps involved are:

1. Scale-space extrema detection
2. Orientation assignment
3. Descriptor extraction

The first step is intended to achieve scale invariance. The second step finds the dominant gradient orientation. All the orientation counts are then made relative to this dominant direction.

For this project following methodology is used:

- In this project the H.O.G(Histogram of Oriented Gradients) is a feature descriptor used in computer vision for image processing for the purpose of object detection.
- HOG features are extracted from large numbers of facial images to be used as part of the recognition mechanism.
- After detecting the faces in the image in the stored student images and from the camera then we encode them.
- After we complete the encoding we compare the faces from the user's camera to the faces of stored in our folder, it will compare them and return the result if it is true it will return the name of that student and we store the student name and the time to a excel file.

Testing

Unit Testing

This type of testing is performed by the developers before the setup is handed over to the testing team to formally execute the test cases. Unit testing is performed by the respective developers on the individual units of source code assigned areas. The developers use test data that is separate from the test data of the quality assurance team.

The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality. In our Food Recipe Web Portal project all the code units are tested properly.

System Testing

The goals of system testing are to detect faults that can only be exposed by testing the entire integrated system or some major part of it. Generally, system testing is mainly concerned with areas such as performance, security, validation, load/stress, and configuration sensitivity. But in our case well focus only on function validation and performance. And in both cases we will use the black-box method of testing.

White Box Testing

In white box testing, the UI is bypassed. Inputs and outputs are tested directly at the code level and the results are compared against specifications. This form of testing ignores the function of the program under test and will focus only on its code and the structure of that code. Test case designers shall generate cases that not only cause each condition to take on all possible values at least once, but that cause each such condition to be executed at least once.

To ensure this happens, we will be applying Branch Testing. Because the functionality of the program is relatively simple, this method will be feasible to apply.

Each function of the binary tree repository is executed independently; therefore, a program flow for each function has been derived from the code.

Black Box Testing

Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would. We have decided to perform Equivalence Partitioning and Boundary Value Analysis testing on our application.

Integration Testing

The testing of combined parts of an application to determine if they function correctly together is Integration testing. There are few methods of doing integration testing.

i) Top-Down.

ii)Bottom-Up.

iii)Sandwich Testing.

In top-down integration testing, the highest-level modules are tested first and progressively lower-level modules are tested after that. In a comprehensive software development environment, bottom-up testing is usually done first, followed by top-down testing. Bottom-Up.

In bottom-up integration testing, the lowest-level modules are tested first and progressively higher-level modules are tested after that.

Whereas the Sandwich testing is a mixed approach of bottom up and top down testing.

Functional System Testing

Regression Testing

Whenever a change in a software application is made it is quite possible that other areas within the application have been affected by this change. To verify that a fixed bug hasn't resulted in another functionality or business rule violation is Regression testing. The intent of Regression testing is to ensure that a change, such as a bug fix did not result in another fault being uncovered in the application. Importance of regression testing:

- Minimize the gaps in testing when an application with changes made has to be tested.
- Testing the new changes to verify that the change made did not affect any other area of the application.
- Test coverage is increased without compromising timelines.
- Increase speed to market the product.

Non Functional System Testing

UI / GUI Testing: Validating if all user interfaces are professionally designed or not is called UI Testing. Through UI testing we verify the followings:

1. Check if all basic elements are available in the page or not.
2. Check spelling of the objects.
3. Check alignments of the objects.
4. Check content displayed in web pages.
5. Check if the mandatory fields are highlights or not.
6. Check consistency in background color, font type and fond size etc.

Installation Testing: checking if we are able to install the software successfully or not as per the guidelines given in installation document.

Un-Installation Testing: checking if we are able to uninstall the software from the system successfully or not.

Compatibility Testing: checking if the application is compatible with the different software and hardware environment.

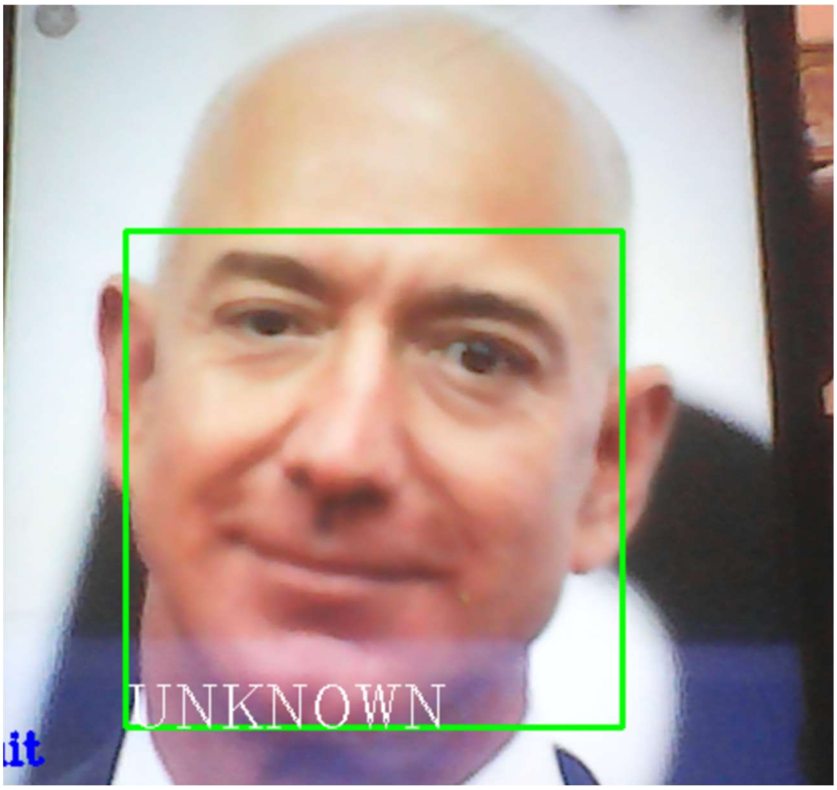
Cross-Browser Testing: checking if the application is running with different browser or not. System Security
There are several issues with respect to system security

1. Authorized: Only authorized users are allowed to access the particular application, which suggests that only logged in user can access the inner pages of this application.
2. If a user types any url to access a page for which he is not authorized then the page will redirect him to Index page.

Test Cases

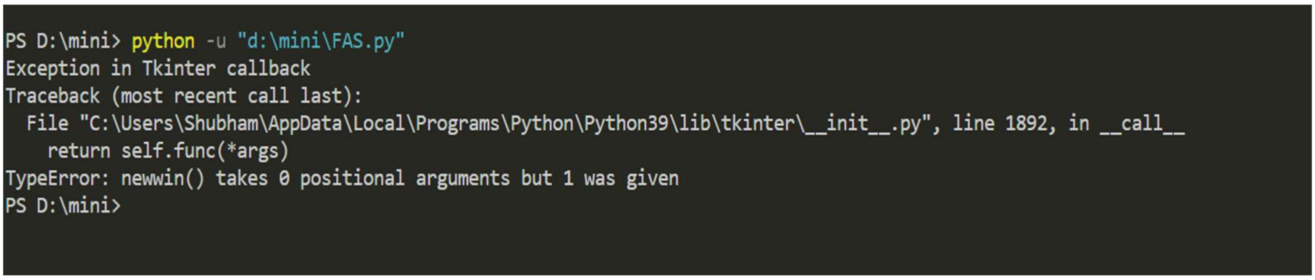
While developing this project we have encountered many bugs/errors these bugs/errors are:

- **Case1**:First problem we encountered while developing this project was that it could not recognize the person which is already stored.



In above image we can see that it is unable to recognize Jeff Bezos. After looking into the code, we finally found that this is happening because we were forgetting to resize the image while encoding.

- **Case2**:Second problem we encountered was it could not create “Start Attendance” window-

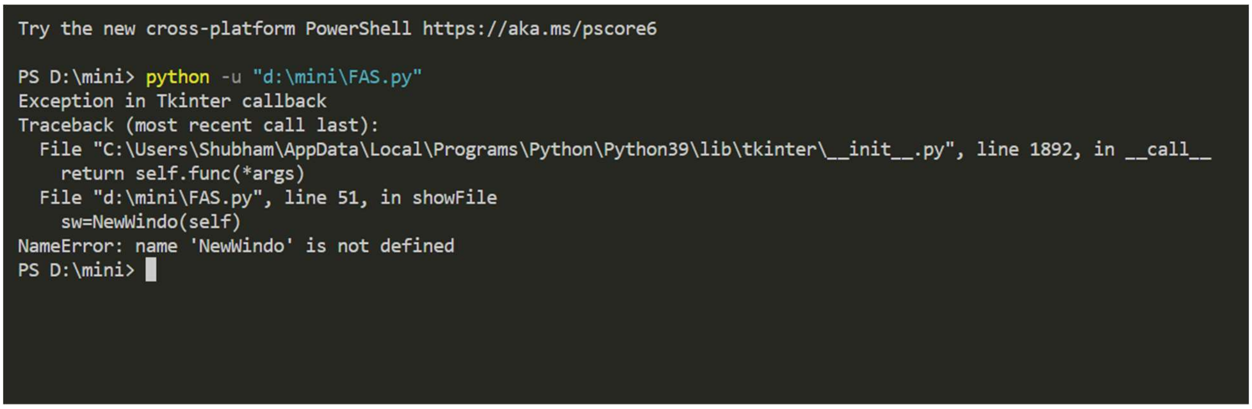


```
PS D:\mini> python -u "d:\mini\FAS.py"
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\Shubham\AppData\Local\Programs\Python\Python39\lib\tkinter\__init__.py", line 1892, in __call__
    return self.func(*args)
TypeError: newwin() takes 0 positional arguments but 1 was given
PS D:\mini>
```

In the above image we can see the problem, this problem occurs because of wrong definition of the method.

- **Case3**:Third problem was

This problem is solved by correctly declaring the object of NewWindow.



```
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\mini> python -u "d:\mini\FAS.py"
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\Shubham\AppData\Local\Programs\Python\Python39\lib\tkinter\__init__.py", line 1892, in __call__
    return self.func(*args)
  File "d:\mini\FAS.py", line 51, in showFile
    sw=NewWindo(self)
NameError: name 'NewWindo' is not defined
PS D:\mini> █
```

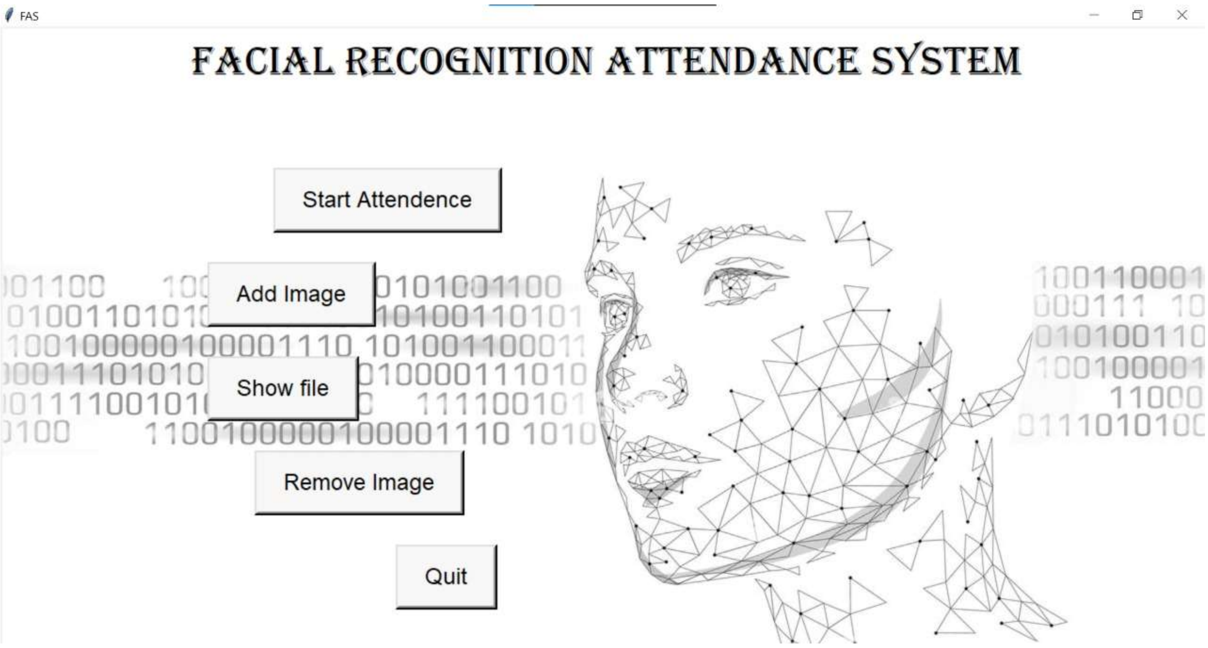
Program Code

```
from tkinter import *
from tkinter import filedialog
import cv2
import face_recognition
import numpy as np
import os,glob
import datetime
from tkinter.messagebox import *
import shutil
import imghdr
import re
from PIL import ImageTk, Image

PATH='images'

class HomeWindow(Tk):


    def __init__(self):
        super().__init__()
        self.fm=FileManager()
        self.images=os.listdir(PATH)
        self.title("FAS")
        self.geometry("1050x620")
        self.bg=ImageTk.PhotoImage(Image.open("Bgimage.png"))
        Label(self,image=self.bg).place(x=0,y=0)#E7F6F2
        Label(text="Facial Recognition Attendance System",font="Algerian 
32",bg="White").place(x=200,y=10)
        Button(text="Start Attendence",font="bold 
18",bg="#F7F7F7",padx=20,pady=10,relief=RAISED,borderwidth=4,command=self.newwin
).place(x=290,y=150)
        Button(text="Add Image",font="bold 
18",bg="#F7F7F7",padx=20,pady=10,relief=RAISED,borderwidth=4,command=self.selctFil
e).place(x=220,y=250)
        Button(text="Show file",font="bold 
18",bg="#F7F7F7",padx=20,pady=10,relief=RAISED,borderwidth=4,command=self.showFi
le).place(x=220,y=350)
        Button(text="Remove Image",font="bold 
18",bg="#F7F7F7",padx=20,pady=10,relief=RAISED,borderwidth=4,command=self.remove
_file).place(x=270,y=450)
        Button(text="Quit",font="bold 
18",bg="#F7F7F7",padx=20,pady=10,relief=RAISED,borderwidth=4,command=quit).place(
x=420,y=550)
        self.state('zoomed')
    def newwin(self):
        if len(self.images)>0:
            nw=NewWindow(self)
            nw.StartWindow()
            nw.grab_set()
        else:
            showwarning(title="Warning",message="No Data Found!")


    def selctFile(self):
        self.fm.fileSelection()
        self.images=os.listdir(PATH)


    def showFile(self):
        if len(self.fm.fileShow())>0:
            sw=NewWindow(self)
            sw.showWindow(self.fm.fileShow())
            sw.grab_set()
        else:
            showinfo(title="Warning",message="No data found!")


    def remove_file(self):
        if len(self.images)>0:
            self.fm.fileDeletion()
            self.images=os.listdir(PATH)
        else:
            showwarning(title="Error",message="No Data Found!")



class NewWindow(Toplevel):


    def __init__(self,parent):
        super().__init__(parent)
        self.regex=re.compile('[.,@_![#$%^&*()<>?/\|}{~:]')
        self.number=['1','2','3','4','5','6','7','8','9','0']
        self.en=StringVar()
        self.box=Listbox(self)


    def detectface(self):
        if len(self.en.get().split()) != 0:
            if self.en.get()[0] not in self.number:
                if not self.regex.search(self.en.get()) and not re.search('[]]', self.en.get()):
                    dect=Detection()
                    dect.FaceDetect(self.en.get())
                    self.destroy()
                    showinfo(title="Attendance Complete",message="Attendence completed
successfully")
                else:
                    showerror("Error",message='Make sure subject name deos not contain any
special character!')
                    self.destroy()
            else:
                showerror("Warning","Subject name can not start with number!")
                self.destroy()
        else:
            showerror(title="Error",message="Please Enter Subject Name")


    def StartWindow(self):
        self.geometry("300x250")
        self.maxsize("300","250")
        Label(self,text="Subject Name",font="bold 12").place(relx = 0.1,rely = 0.2)
        Entry(self,textvariable=self.en,font="bold 10").place(relx = 0.45,rely = 0.22)
        Button(self,text="Start",font="bold
12",command=self.detectface).pack(anchor=CENTER,pady=90)


    def showWindow(self,files):
        self.geometry("200x180")
        self.maxsize("200","180")
        self.box=Listbox(self,height = 10,width = 35,font="bold 10")
        scrollbar= Scrollbar(self, orient= 'vertical')
        scrollbar.pack(side= RIGHT, fill= BOTH)
        self.box.pack()
        Scrollbar(self.box).pack
        for file in files:
            self.box.insert(END,file)
        self.box.config(yscrollcommand= scrollbar.set)
        scrollbar.config(command= self.box.yview)
        self.box.bind("<Double 1>", self.curserSelection)


    def curserSelection(self,event):
        x = self.box.curselection()[0]
        file = self.box.get(x)
        f=FileManager()
        f.stratFile(file)



class Detection():   # detection and attendance
```

```
    def __init__(self):
        self.names=[]
        self.time=datetime.datetime.now()
        self.encodedlist=self.encoding()
        self.cap = cv2.VideoCapture(0)


    def FaceDetect(self,en):
        while(True):
        # Capture frame-by-frame
            ret, frame = self.cap.read()
            faces=face_recognition.face_locations(cv2.resize(frame, (0,0), None, 0.5,0.5))  #
model hog by default
            encode_current_image=face_recognition.face_encodings(cv2.resize(frame, (0,0),
None, 0.5,0.5),faces)
            for encoded_face,faceL in zip(encode_current_image,faces):
                top, right, bottom, left = faceL
                top*=2
                right*=2
                bottom*=2
                left*=2
                cv2.rectangle(frame, (left, top), (right, bottom), (0, 255, 0), 2)
                try:
                    matchface=face_recognition.compare_faces(self.encodedlist, encoded_face)
                    result=face_recognition.face_distance(self.encodedlist, encoded_face)
                    index=np.argmin(result)
                    if matchface[index]:
                        name=self.names[index].upper()
                        self.takeAtttendance(name,en)

cv2.putText(frame,f"{name}",(left+5,top),cv2.FONT_HERSHEY_COMPLEX,0.5,(255,255,
255),1)
                        cv2.putText(frame,f"{name}'s attendance
marked",(200,50),cv2.FONT_HERSHEY_COMPLEX,0.5,(255,255,255),1)
                    else:

cv2.putText(frame,"UNKNOWN",(left,bottom),cv2.FONT_HERSHEY_COMPLEX,1,(255,
255,255),1)
                # cv2.rectangle(frame, (left, bottom-35), (right, bottom), (0, 255, 0), cv2.FILLED)
                except:

cv2.putText(frame,"UNKNOWN",(left,bottom),cv2.FONT_HERSHEY_COMPLEX,1,(255,
255,255),1)
            cv2.putText(frame,"Found {0}
faces!".format(len(faces)),(10,30),cv2.FONT_HERSHEY_COMPLEX,0.7,(255,0,0),2)
            cv2.putText(frame,"Press Q to
quit",(10,450),cv2.FONT_HERSHEY_COMPLEX,0.7,(255,0,0),2)
```

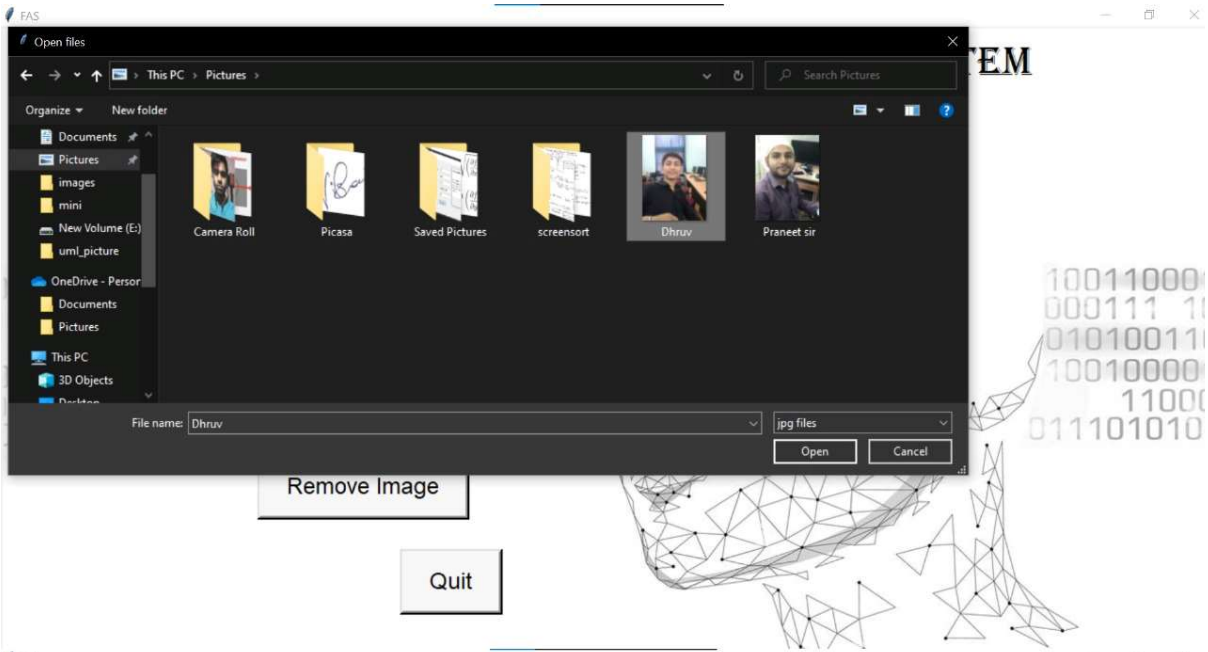
```
            # Display the resulting frame
            cv2.imshow('Attendence', frame)
            if cv2.waitKey(1) & 0xFF == ord('q'):
                break
        self.cap.release()
        cv2.destroyAllWindows()


    def encoding(self):
        ername=""
        encode=[]
        imageList=[]
        self.names.clear()
        images=os.listdir(PATH)
        for cl in images:
            pimg=cv2.imread(f"{PATH}/{cl}")
            imageList.append(pimg)
            self.names.append(os.path.splitext(cl)[0])
        for img,nm in zip(imageList,self.names):
            try:
                image=cv2.resize(img, (0,0), None, 0.25,0.25)
                encoded_face = face_recognition.face_encodings(image)[0]
                encode.append(encoded_face)
            except:
                # print(nm)
                ername=ername+" "+nm
        if ername:
            showwarning(message=f"{ername} is not clear or no face detected")
        return encode


    def takeAtttendance(self,name,en):
        try:
            with open(f"{en.replace(' ',")} {self.time.day}-{self.time.month}.csv","r") as f:
                if name not in f.read():
                    with open(f"{en.replace(' ', ")} {self.time.day}-{self.time.month}.csv","a") as f2:
                        f2.write(f"\n{name},{self.time.hour}:{self.time.minute}")
        except FileNotFoundError:
            with open(f"{en.replace(' ',")} {self.time.day}-{self.time.month}.csv","w") as f:
                f.write("Name,Time")
                f.write(f"\n{name},{self.time.hour}:{self.time.minute}")


class FileManager():
    def __init__(self):
        self.file=(('jpg files','*.jpg*'),('All files', '*.*'))
        self.name=""
        self.f=Detection()
    def fileSelection(self):
        self.name=""
        filenames=filedialog.askopenfilenames(title='Open files',initialdir='/',filetypes=self.file)
        if filenames:
            for filename in filenames:
                if imghdr.what(filename):
                    if not os.path.exists(PATH+"\\"+os.path.basename(filename)):
                        shutil.copy(filename,PATH)
                        self.name=self.name+os.path.basename(filename)+","
                    else:
                        showinfo(message="File already exist",title="Existing Data")
                else:
                    showwarning(title="Warning",message="Please select images")
            if self.name:
                self.f.encoding()
                showinfo(title="Image Addition",message=f"{self.name[:-1]} added")


    def fileShow(self):
        files=glob.glob("*.csv")
        return files


    def stratFile(self,file):
        os.startfile(file)


    def fileDeletion(self):
        name=""
        filet=(('jpg files','*.jpg*'),('All files', '*.*'))
        filenames=filedialog.askopenfilenames(title='Open
files',initialdir=PATH,filetypes=filet)
        if filenames:
            for filename in filenames:
                if os.getcwd()+"\\"+PATH == os.path.dirname(filename).replace("/","\\"):
                    os.remove(filename)
                    name+=os.path.basename(filename)+","
                else:
                    showerror(message="Please select image from the current folder",title="Out of
range")
            if name:
                showinfo(message=f"{name[:-1]} deleted",title="Image Deletion")

if __name__=="__main__":
    if PATH not in os.listdir():
        os.mkdir(PATH)
    hw=HomeWindow()
    hw.mainloop()
```

Snapshot of project



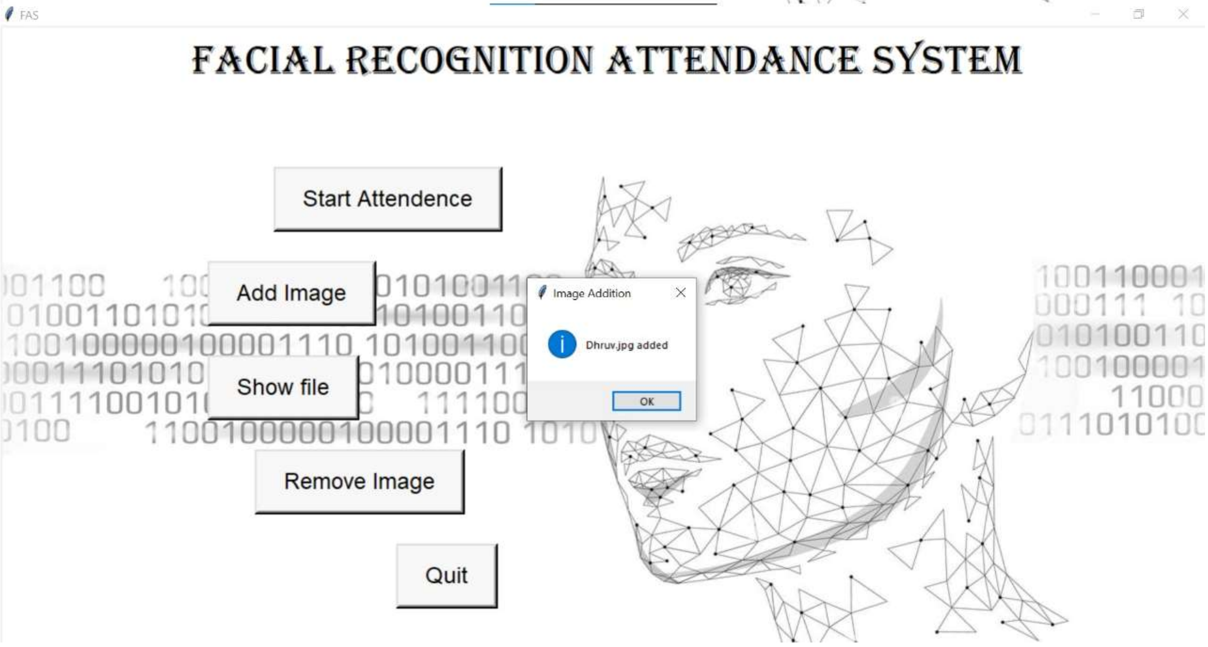
Interface of the project

Adding image



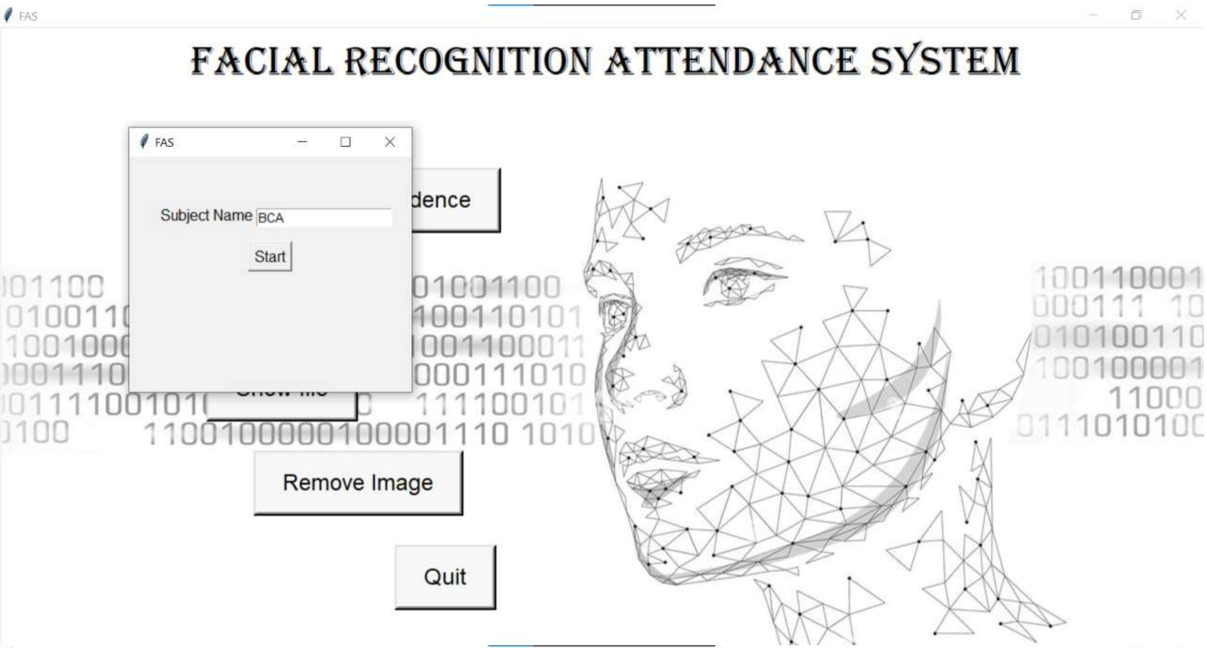
File name: Dhruv

jpg files

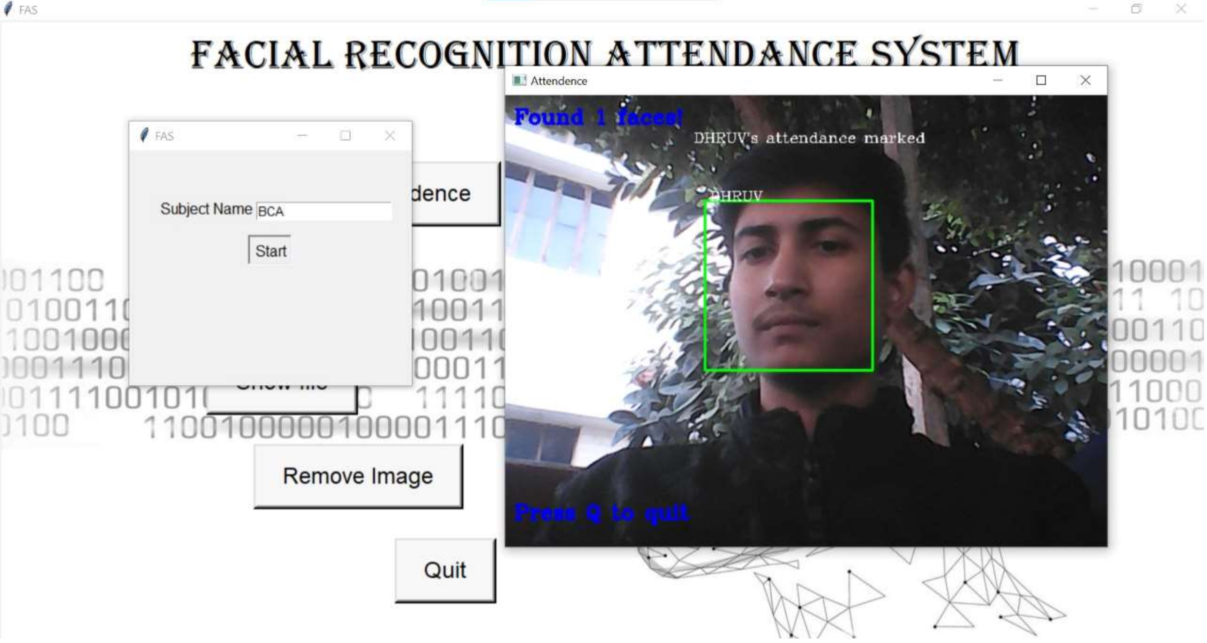


Dhruv.jpg added

Starting Attendance

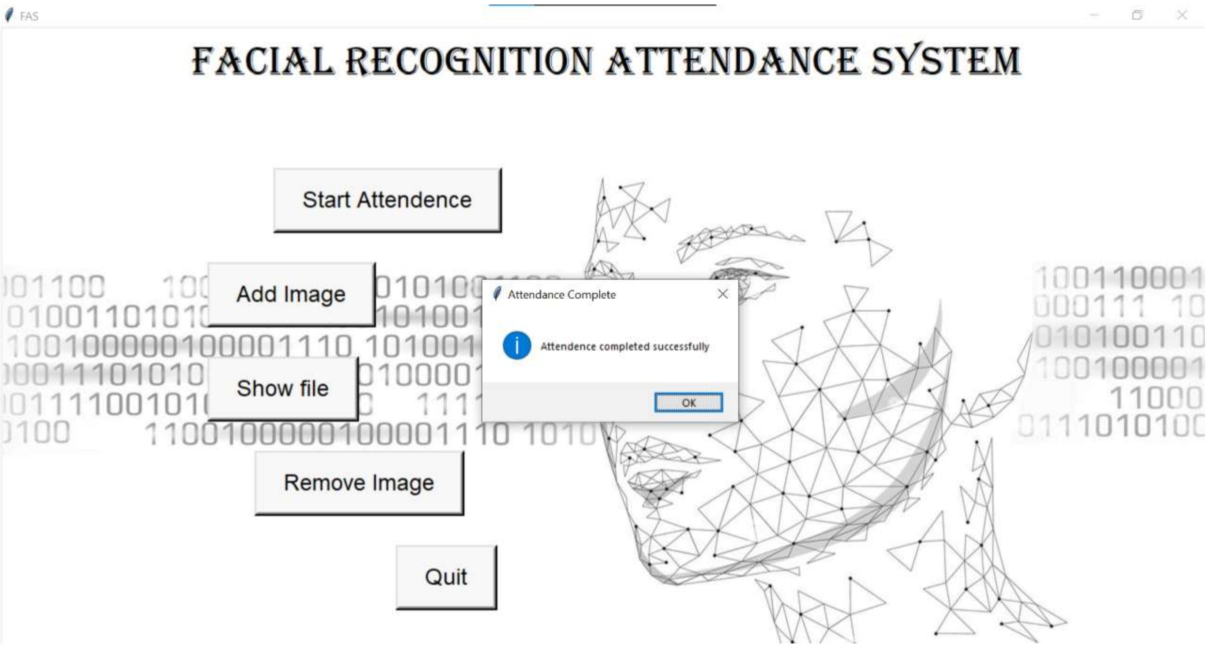


Subject Name BCA



Subject Name BCA

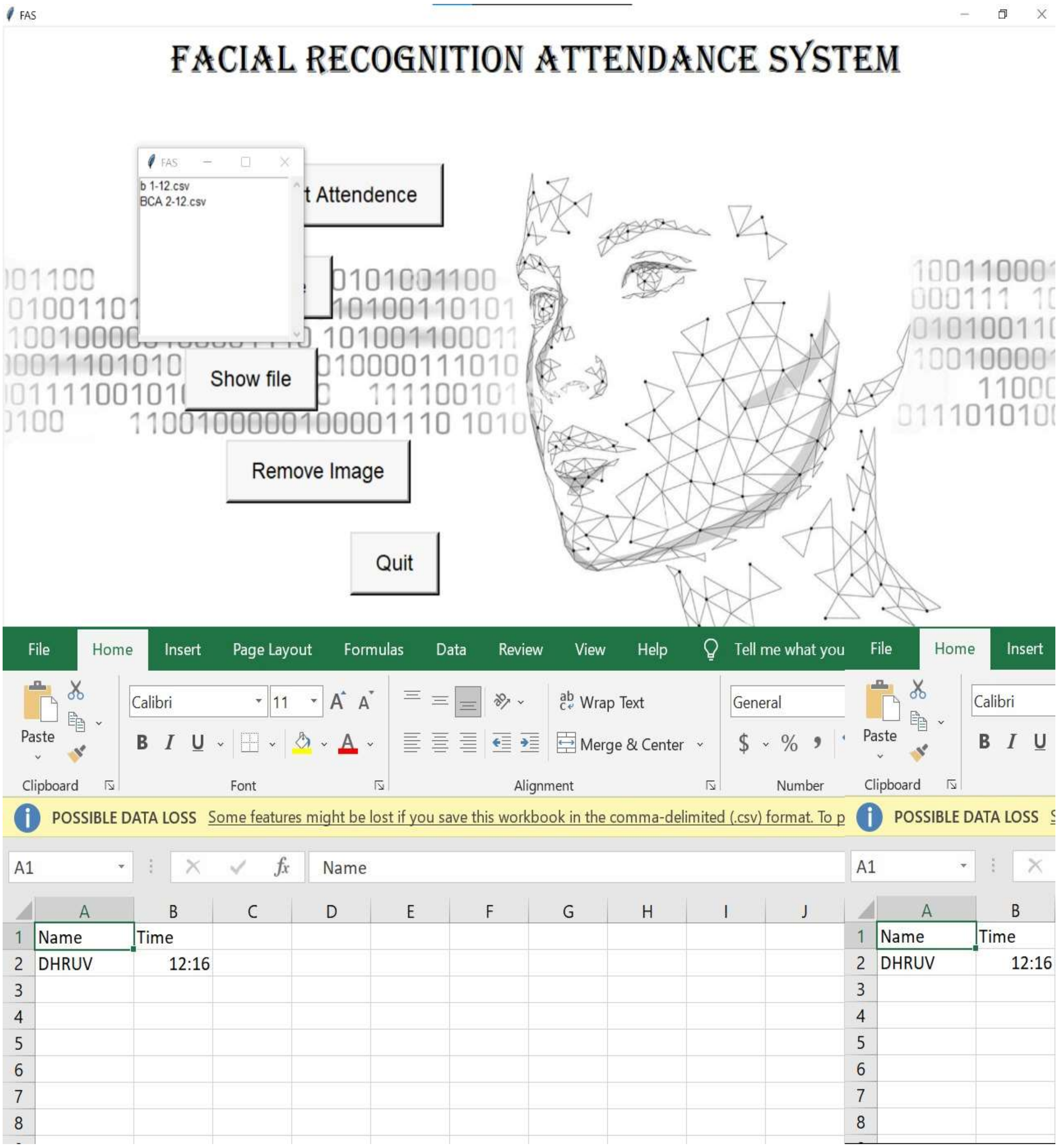
```
Found 1 faces!
DHRUV's attendance marked
DHRUV
Press Q to quit
```



Attendance Complete

Attendence completed successfully

Showing the attendance file



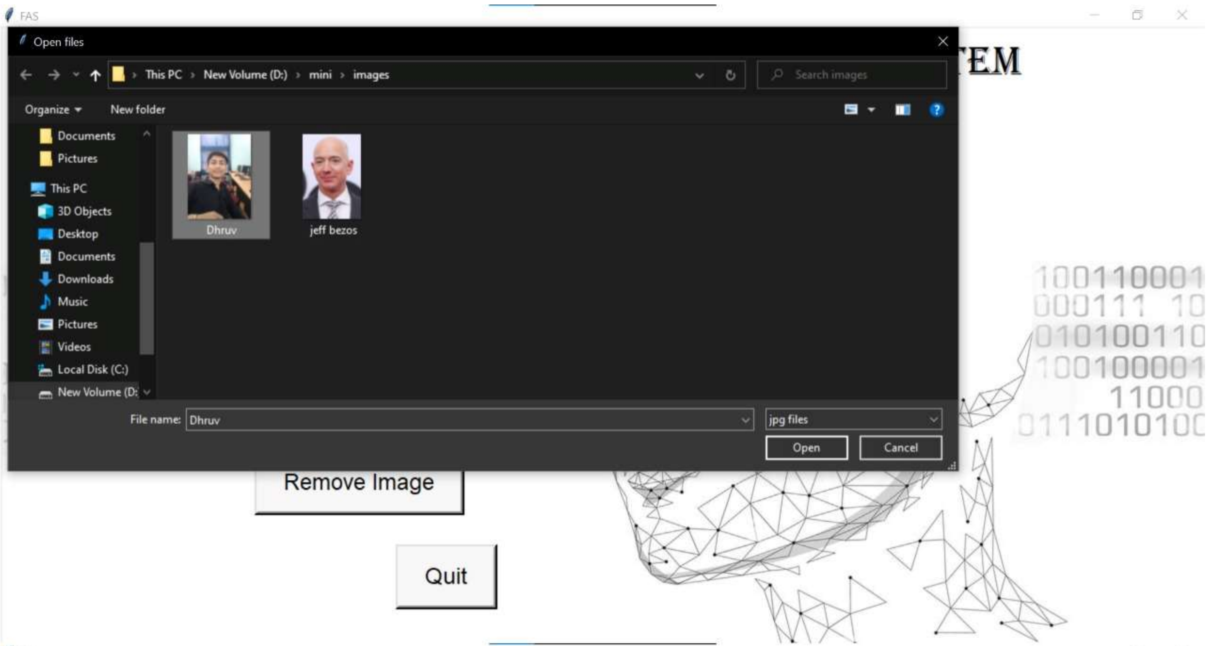
POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To p

	A	B	C	D	E	F	G	H	I	J
1	Name	Time								
2	DHRUV	12:16								
3										
4										
5										
6										
7										
8										

POSSIBLE DATA LOSS S

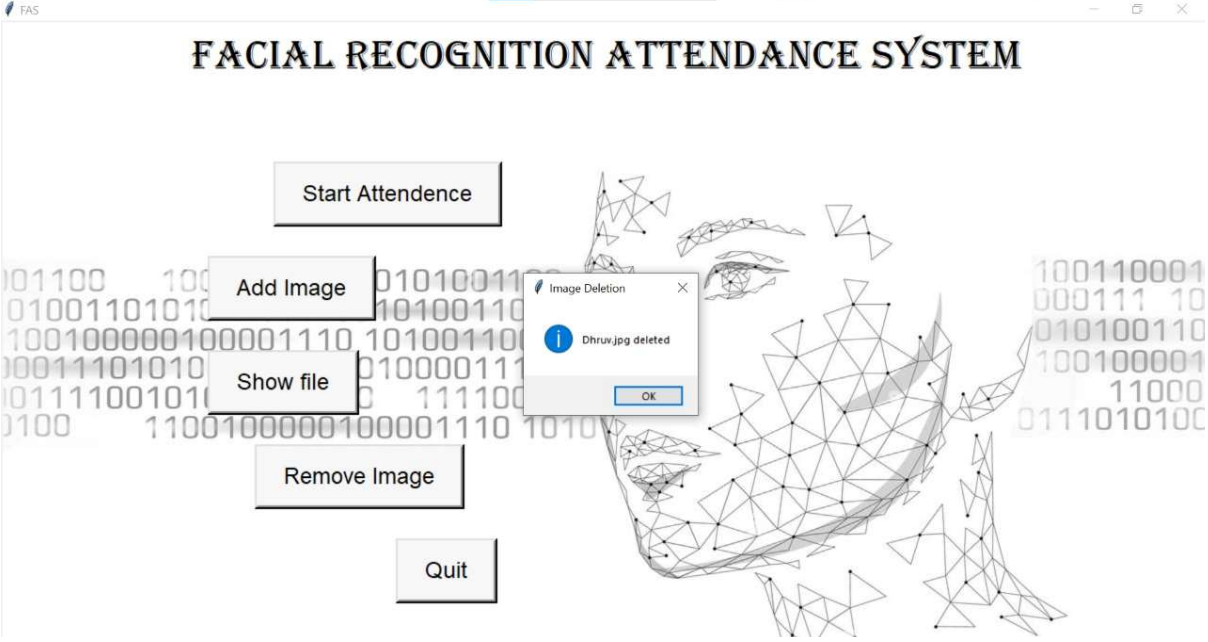
	A	B
1	Name	Time
2	DHRUV	12:16
3		
4		
5		
6		
7		
8		

Removing the image



File name: Dhruv

jpg files



Dhruv.jpg deleted

Future Scope

In a nutshell, it can be summarized that the future scope of this project circles around maintaining information regarding ~

- Simplicity: We will make the Application more simple so as no third party would be needed to explain how to work it.

- Learning: The upcoming generation shall see these types of changes in regular day by day procedure of attendance so as more students built a keen to learn the new technology and develop in mean time.

- Accuracy: The main problem faced now in this project is accuracy to identify faces 100%. So in upcoming time more accurate processor or some type of machinery maybe built for accurate comparison and identification.

- Availability: The availability of the project to a specified working place or institution for fast and smooth management of attendance.

- Data Collectivity: The data collected at end of a session or a semester shall be organised and presented in a single file after combining all the separate day by day files.

- Backup: we can add a backup mechanism for the data or the file to be lost can be retrieved again.

CONCLUSION

Before the development of this project. There were many loopholes in the process of taking attendance using the old method which caused many troubles to most of the institutions. Therefore, the facial recognition feature embedded in the attendance monitoring system can not only ensure attendance to be taken accurately and also eliminated the flaws in the previous system. By using technology to conquer the defects cannot merely save resources but also reduces human intervention in the whole process by handling all the complicated task to the machine. The only cost to this solution is to have sufficient space in to store all the faces into the database storage. Fortunately, there is such existence of micro SD that can compensate with the volume of the data. In this project, the face database is successfully built. Apart from that, the face recognizing system is also working well.

At the end, the system not only resolve troubles that exist in the old model but also provide convenience to the user to access the information collected by mailing the attendance sheet to the respected faculty.

Reference

- **www.towarddatascience.com**
- **www.python.org**
- **www.geekforgeeks.com**
- **www.javatpoint.com**
- **www.stackoverflow.com**