CHAPTER 1

INTRODUCTION

Maintaining the attendance is very important in all the institutes for checking the performance of employees . Every institute has its own 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. But in these methods employees have to wait for long time in making a queue at time they enter the office. Many biometric systems are available but the key authentications are same is all the techniques.

Every biometric system consists of enrolment process in which unique features of a person is stored in the database and then there are processes of identification and verification. These two processes compare the biometric feature of a person with previously stored template captured at the time of enrollment. Biometric templates can be of many types like Fingerprints, Eye Iris, Face, Hand Geometry, Signature, Gait and voice. Our system uses the face recognition approach for the automatic attendance of employees in the office room environment without employees' intervention . Face recognition consists of two steps, in first step faces are detected in the image and then these detected faces are compared with the database for verification. A number of methods have been proposed for face detection i.e. Ada Boost algorithm, the Float Boost algorithm, the S-Ada Boost algorithm Support Vector Machines (SVM), and the Bayes classifier. The efficiency of face recognition algorithm can be increased with the fast face d tection algorithm. In all the above methods SURF is most efficient. Our system utilized this algorithm for the detection of faces in the office room image. Face recognition techniques can be Divided into two types Appearance based which use texture features that is applied to whole face or some specific Regions, other is Feature based which uses geometric features like mouth, nose, eyes, eye brows, cheeks and Relation between them. Statistical tools such as Linear Discriminant Analysis (LDA), Principal Component Analysis (PCA), Kernel Methods, and Neural Networks, Eigen-faces have been used for construction of face templates. Illumination invariant algorithm is utilized for removing the lighting effect inside the office room.

1.1 OBJECTIVE& SCOPE

- The main objective of this project is to offer system that simplify and automate the process of recording and tracking students' attendance through face recognition technology. It is biometric technology to identify or verify a person from a digital image or surveillance video.

- Face recognition is widely used nowadays in different areas such as universities, banks, airports, and offices. We will use preprocessing techniques to detect, recognize and verify the captured faces like Eigen faces method. We aim to provide a system that will make the attendance process faster and more precisely.

- To develop a software that will facilitate attendance management by detection and recognition of students faces and record the attendance in attendance.csv live using webcam

- The system would be able to match detected students faces cropped from an image to those on a database on the system

1.2 PURPOSE OF THE PROJECT

Automatic face recognition (AFR) technologies have seen dramatic improvements in performance over the past years, and such systems are now widely used for security and commercial applications. An automated system for human face recognition in a real time background for a college to mark the attendance of their employees. So Smart Attendance using Real Time Face Recognition is a real world solution which comes with day to day activities of handling employees. The task is very difficult as the real time background subtraction in an image is still a challenge . To detect real time human face are used and a simple fast Principal Component Analysis has used to recognize the faces detected with a high accuracy rate. The matched face is used to mark attendance of the employee.Our system maintains the attendance records of employees automatically. Manual entering of attendance in logbooks becomes a difficult task and it also wastes the time. So we designed an efficient module that comprises of face recognition to ma age the attendance records of employees. Our module enrols the staff's face . This enrolling is a onetime process and their face will be stored in the database. During enrolling of face we require a system since it is a onetime process. You can have your own roll

number as your employee id which will be unique for each employee. The presence of each employee will be updated in a database. The results showed improved performance over manual attendance management system. Attendance is marked after employee identification. This product gives much more solutions with accurate results in user interactive manner rather than existing attendance and leave management systems.

1.3 METHODOLOGY

Agile is a set of techniques to manage software development projects. It consists in: • Being able to respond to changes and new requirements quickly. • Teamwork, even with the client. • Building operating software over extensive documentation. • Individuals and their interaction over tools. We believed it was a perfect fit for our project since we did not know most requirements beforehand. By using the Agile, we were able to focus only on the features which had the most priority at the time.

- **Agile project management** is an iterative approach to **project management** which allows you to break large projects down into more manageable tasks tackled in short iterations or sprints. This enables your team to adapt to change quickly and deliver work fast.

FIG 1 (**Agile Methodology**)

3

1.4 Project time line

Sr. no.	Task	Date
1.	Installations of Pycharm ((python environment) along with OpenCV libraries	2 Sept
2.	Conduct a literature survey to identify the algorithms suitable for developing our model	8 Sept
3.	Developing Design and Architecture of our model, by resource implementation and initial prototyping	27 Sept
4.	Implementing necessary changes with given feedback to counter underlying problems and tackle necessities	10 Oct
5.	Review and Evaluation of our model with full-scale testing and corrections	19 Oct
6.	System testing to improve the features	5 Nov
7.	Complete documentation of the project with details in the Project Report	10 Nov

1.5 Gantt Chart

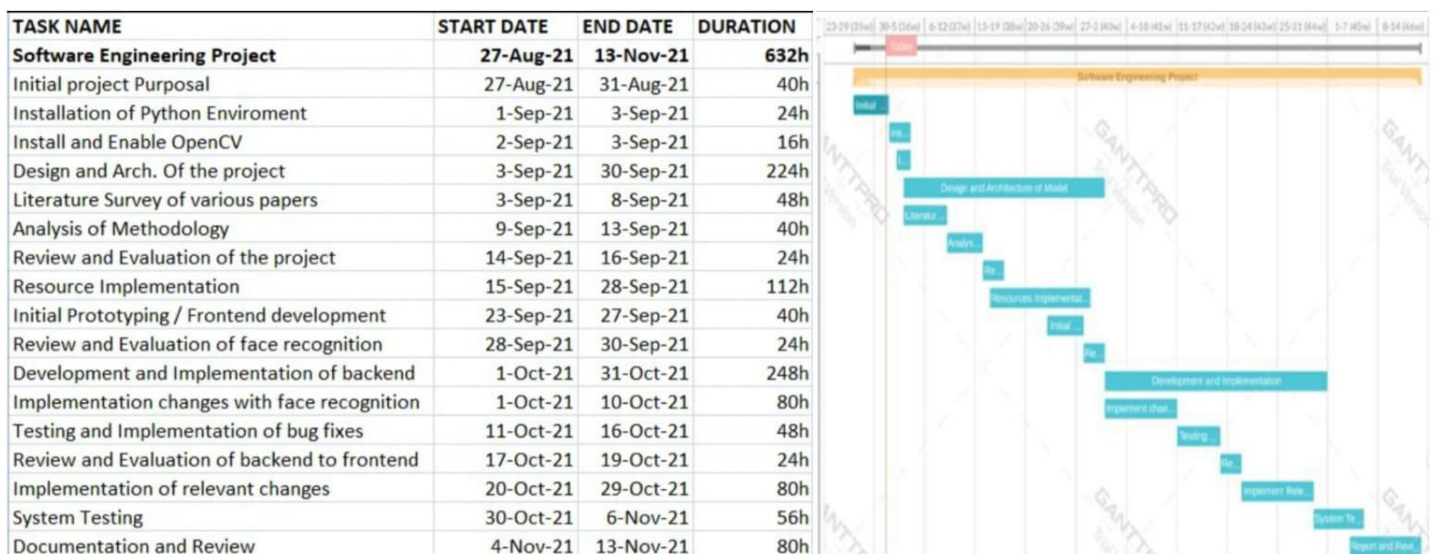| TASK NAME	START DATE	END DATE	DURATION
Software Engineering Project	27-Aug-21	13-Nov-21	632h
Initial project Purposal	27-Aug-21	31-Aug-21	40h
Installation of Python Enviroment	1-Sep-21	3-Sep-21	24h
Install and Enable OpenCV	2-Sep-21	3-Sep-21	16h
Design and Arch. Of the project	3-Sep-21	30-Sep-21	224h
Literature Survey of various papers	3-Sep-21	8-Sep-21	48h
Analysis of Methodology	9-Sep-21	13-Sep-21	40h
Review and Evaluation of the project	14-Sep-21	16-Sep-21	24h
Resource Implementation	15-Sep-21	28-Sep-21	112h
Initial Prototyping / Frontend development	23-Sep-21	27-Sep-21	40h
Review and Evaluation of face recognition	28-Sep-21	30-Sep-21	24h
Development and Implementation of backend	1-Oct-21	31-Oct-21	248h
Implementation changes with face recognition	1-Oct-21	10-Oct-21	80h
Testing and Implementation of bug fixes	11-Oct-21	16-Oct-21	48h
Review and Evaluation of backend to frontend	17-Oct-21	19-Oct-21	24h
Implementation of relevant changes	20-Oct-21	29-Oct-21	80h
System Testing	30-Oct-21	6-Nov-21	56h
Documentation and Review	4-Nov-21	13-Nov-21	80h

FIG 2(Gantt Chart)

1.6 <u>TECHNICAL DETAILS</u>

<u>SOFTWARE REQUIREMENTS:</u>

- Python == 3.6.7
- tkinter==1.3.3, os-sys==2.1.4 , six==1.16.0(Front End)
- numpy == 1.21.2 , opencv-python==4.5.3.56 , pa https://pandas.pydata.org/ndas==1.3.2 , pillows==8.3.2 , python-datautil==2.8.2 (Back End)
- Microsoft Windows , Linux , Mac (All OF THEM)
- Internet Explorer (5.0 and above) or Mozilla Firefox (6.0 and above)
- VSCODE text editor , A Bash terminal / command prompt.

<u>HARDWARE REQUIREMENTS</u>:
- Intel core 2 (minimum spec)
- Ram :- 2 GB
- Computer Memory :- 128 GB
- Intel HD graphics (minimum version 15.40.46.5144)

1.7 <u>ADVANTAGES AND DISADVANTAGES :-</u>

- **<u>ADVANTAGES</u>**

Automated time tracking system:-

Offices or workplaces or even just public places where the entry and exit times of employees or a person are strictly noted down will have a ready-made automated system to record the entry and exit time of each person for a given time. It won't even need the person to stop and click a photo, the software's are advanced enough to record the data from a continuous reel also. This means the flow won't get hampered, or you won't have to stop and smile or something like that. Just enter or exit the place effortlessly like you do everyday and boom! Your attendance will be recorded without any fuss!.

Cost-effective:-

Since the whole process will be done by a computer, it means the total attendance registration and calculation will be automated and done by the system itself, therefore, saving us the money which would have been otherwise spent on the labor cost to do that.

Increased security:-

Face recognition based attendance system won't just calculate attendance but also note down the entry and exits of visitors in the place. At times when there is a situation where the identity and time of entry and exit of a specific person need to be noted, this system would become handy as it will easily show you when he/she came in and what are the places he/she went to a very precise level. All of this means, you will have a much higher security level in your workplace.

Time saving:-

The whole world is suffering from COVID19 and it is high time we must give heed to social distancing. Having a safe distance with others has become a necessity nowadays. Times like this can be problematic if you have manual attendance system, Having a Face recognition based attendance system will not only allow you to register the attendance of a person but also keep you at a safe distance from them as you can work remotely and still see who all are coming and going. This calls for the point that, this whole system is a much safer, time-saving, and faster method to record attendance.

Easy to manage:-

Since the artificial intelligence based attendance system is fully automated, managing the records and keeping a track of day to day activities will become much easier than the manual system. Everything will be done by the system. Many software are programmed in such a way that it shows the exact time of how many hours or minutes a person worked on his/her desk in the day. All this is can be done on a very large scale. Just imagine, recording the activities of a large crowd of 200 people simultaneously without any fuss and recording it at the same time in an organized manner!. Such is the power of AI in face recognition.

- **DISADVANTAGE**

If in the wrong hands, it will be a disaster:-

Just like we have already read that, the touchless biometric attendance system will record everything and is a handy method to analyze and register a person's activities, if in wrong hands can cause a serious level of the privacy breach. Governments in some of the most treacherous countries in the world use this technology to analyze the activities of their citizens. Which means you won't be able to go anywhere without the government knowing. Cases, where a particular person or a group uses contactless biometric attendance, will mean, they can easily monitor the whereabouts of a person meaning you will be tracked at all times. This means that AI in Face recognition system can be easily worked against us.

Data privacy breach:-

Data in these systems means, billions of pictures and god knows, 1000s of hours of video footage saved into a hard drive. This calls for the point when this hard drive gets hacked or smuggled to get data of a particular person. The data can be easily tampered and used to in against a particular organization or person. Which can be very much dangerous. Therefore, companies that use Artificial intelligence based attendance system have a very strict security to safeguard the information of their employees**.**

Low reliability:-

Sometimes there have been instances where the identity of a person is not able to get verified. There have been also cases where the identity of a person is verified with another person's identity. This means a person who is "X" is recognized as "Y" instead of "X". Times like this can be misused as criminals can loot or commit a crime in the name of other people. This means, that even though we have achieved a lot of advancement in the technology sector, there will always be a small gap between us and the ideal system.

Lack of regulations in the AI in Face recognition systems:-

We can already guess how a particular Artificial intelligence based attendance system can be misused. In addition to all these, the govt across the whole world don't have a particular policy or regulations to regulate the usage of touch less biometric attendance system. Some cities and places in the world have already banned this system considering the amount of threat it can cause

to the people. While there are some places in the world where usage of face recognition based attendance system is already under progress but without any regulation. Which means lakhs of people's data is already under threat.

Keeping all these things in mind, the advantages of Artificial intelligence based attendance system are many, but so are the disadvantages. But we can see that this system will be much better if all the processes are regulated and taken care of. The first point will be:-

Proper rules and regulations while setting up and using touchless biometric attendance system. Companies or organizations should be warned on what level the information of people can be used or shared. Keeping the sharing freedom to the least extent possible, the government can easily consult experts and set up policies particularly based on this Artificial intelligence based attendance systems.

Secondly, companies or organizations should pay much attention to the security of the information rather than other things in setting up this system. The major setup that this whole contactless biometric attendance will need is a small camera at the entry and exit points of the rooms and a computer and a storage device to store the data. All this won't take more than $2\times2''$ space. So, keeping these things aside, people should be much more advanced and strict when it comes to the security of this system.

Thirdly, face recognition based attendance system should be worked more and more every day to take it closer to the ideal system that we need. The algorithms should be worked upon and data set that is feed initially to the system should be much larger to remove glitches. Keeping in mind, the glitches that this system can cause can be much disastrous in situations where a person gets paid on a "no work no pay" policy job. He/she may get marked absent on a particular day even if he had come, just because the contactless biometric attendance system was faulty and not accurate enough to register his face. At police stations or courts where the identity of a person is identified using this technology, it may cause some serious trouble if a particular person is identified as someone else in the records. This will not only create a problem but also allow criminals to make use of this flaw every time they commit a crime.

CHAPTER 2

IMPLEMENTATION / TECHNOLOGY ENVIRONMENT

2.1 Explanation:

The system consists of a camera that captures the images of the employee and sends it to the image enhancement module. After enhancement the image comes in the Face Detection and Recognition modules and then the attendance is marked on the database server. This is shown in the experimental setup in Figure. At the time of enrolment, templates of face images of individual employees are stored in the Face database. Here all the faces are detected from the input image and the algorithm compares them one by one with the face database. If any face is recognized the attendance is marked on the server from where anyone can access and use it for different purposes. In this way a lot of time is saved and this is highly secure process no one can mark the attendance of other. Attendance is maintained on the server so anyone can access it for purposes like administration, employees themselves. In order to avoid the false detection we are using the skin classification technique. Using this technique enhance the efficiency and accuracy of the detection process. In this process first the skin is classified and then only skin pixels remains and all other pixels in the image are set to black, this greatly enhance the accuracy of face detection process. Two databases are displayed in the experimental setup. Face Database is the collection of face images and extracted features at the time of enrolment process and the second attendance database contains the information about the employees and also uses to mark attendance.

2.2 Tools and Technology used

FRONTEND

- We are using TKINTER and OS for the frontend part and operating system interfaces.
- We are using PYTHON-DATEUTIL for system data identification.
- We are using PYTHON-TIME for system time identification.

BACKEND

- We are using OPENCV for the computer vision, machine learning and image processing.
- We are using NUMPY for mathematical calculation.
- We are using PANDAS for the operations for manipulating numerical data and time series.
- We are using PIL for lightweight image processing that aids in editing, creating and saving images.
- We are using python CSV for creating a comma separated files to analyze data.
- And we are using HAAR CASCADE classification for object and image detection.

2.3 TKINTER

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

To create a tkinter app:

Importing the module – tkinter

Create the main window (container)

Add any number of widgets to the main window

Apply the event Trigger on the widgets.

2.4 OS

The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules. This module provides a portable way of using operating system-dependent functionality. The *os* and *os.path* modules include many functions to interact with the file system.The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules. This module provides a portable way of using operating system-dependent functionality. The *os* and *os.path* modules include many functions to interact with the file system.

2.5 <u>PYTHON TIME</u>

Time module in Python provides various time-related functions. This module comes under Python's standard utility modules.

time.time() method of Time module is used to get the time in seconds since epoch. The handling of leap seconds is platform dependent.

Note: The epoch is the point where the time starts, and is platform dependent. On Windows and most Unix systems, the epoch is January 1, 1970, 00:00:00 (UTC) and leap seconds are not counted towards the time in seconds since the epoch. To check what the epoch is on a given platform we can use time.gmtime(0).

2.6 <u>PYTHON DATE-UTILS</u>

The parser module can parse datetime strings in many more formats. There can be no better library than dateutil to parse dates and times in Python. To lookup the timezones, the tz module provides everything. When these modules are combined, they make it very easy to parse strings into timezone-aware datetime objects.

2.7 <u>OPENCV</u>

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.

This OpenCV tutorial will help you learn the Image-processing from Basics to Advance, like operations on Images, Videos using a huge set of Opencv-programs and projects.

2.8 <u>NUMPY</u>

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python.

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data.

2.9 PANDAS

Pandas is an open-source library that is built on top of NumPy library. It is a Python package that offers various data structures and operations for manipulating numerical data and time series. It is mainly popular for importing and analyzing data much easier. Pandas is fast and it has high-performance & productivity for users.

2.10 PILLOW (a fork of PIL)

Python Imaging Library (expansion of PIL) is the de facto image processing package for Python language. It incorporates lightweight image processing tools that aids in editing, creating and saving images. Support for Python Imaging Library got discontinued in 2011, but a project named pillow forked the original PIL project and added Python3.x support to it. Pillow was announced as a replacement for PIL for future usage. Pillow supports a large number of image file formats including BMP, PNG, JPEG, and TIFF. The library encourages adding support for newer formats in the library by creating new file decoders.

2.11 PYTHON CSV

CSV (Comma Separated Values) is a simple file format used to store tabular data, such as a spreadsheet or database. A CSV file stores tabular data (numbers and text) in plain text. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format.

2.12 HAARCASCADE

Haar Cascade classifiers are an effective way for object detection. This method was proposed by Paul Viola and Michael Jones in their paper Rapid Object Detection using a Boosted Cascade of Simple Features .Haar Cascade is a machine learning-based approach where a lot of positive and negative images are used to train the classifier.

Positive images – These images contain the images which we want our classifier to identify.

Negative Images – Images of everything else, which do not contain the object we want to detect.

2.13 Local Binary Patterns Histograms (LBPH).

Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number.

It was first described in 1994 (LBP) and has since been found to be a powerful feature for texture classification. It has further been determined that when LBP is combined with histograms of oriented gradients (HOG) descriptor, it improves the detection performance considerably on some datasets.Using the LBP combined with histograms we can represent the face images with a simple data vector.

The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. To do so, the algorithm uses a concept of a sliding window, based on the parameters radius and neighbors and hence it compute the identification of object and image surrounding.

2.14 TECHNOLOGY ENVIRONMENT

2.15 Introduction to Visual Studio

Visual Studio is an Integrated Development Environment(IDE) developed by Microsoft to develop GUI(Graphical User Interface), console, Web applications, web apps, mobile apps, cloud, and web services, etc. With the help of this IDE, you can create managed code as well as native code. It uses the various platforms of Microsoft software development software like Windows store, Microsoft Silverlight, and Windows API, etc. It is not a language-specific IDE as you can use this to write code in C#, C++, VB(Visual Basic), Python, JavaScript, and many more languages. It provides support for 36 different programming languages. It is available for Windows as well as for macOS.

Evolution of Visual Studio: The first version of VS(Visual Studio) was released in 1997, named as Visual Studio 97 having version number 5.0. The latest version of Visual Studio is

15.0 which was released on March 7, 2017. It is also termed as Visual Studio 2017. The supported .Net Framework Versions in latest Visual Studio is 3.5 to 4.7. Java was supported in old versions of Visual Studio but in the latest version doesn't provide any support for Java language.

Visual Studio Editions

There are 3 editions of Microsoft Visual Studio as follows:

Community: It is a free version which is announced in 2014. All other editions are paid. This contains the features similar to Professional edition. Using this edition, any individual developer can develop their own free or paid apps like .Net applications, Web applications and many more. In an enterprise organization, this edition has some limitations. For example, if your organization have more than 250 PCs and having annual revenue greater than $1 Million(US Dollars) then you are not permitted to use this edition. In a non-enterprise organization, up to five users can use this edition. Its main purpose is to provide the Ecosystem(Access to thousands of extensions) and Languages(You can code in C#, VB, F#, C++, HTML, JavaScript, Python, etc.) support.

 Professional: It is the commercial edition of Visual Studio. It comes in Visual Studio 2010 and later versions. It provides the support for XML and XSLT editing and includes the tool like Server Explorer and integration with Microsoft SQL Server. Microsoft provides a free trial of this edition and after the trial period, the user has to pay to continue using it. Its main purpose is to provide Flexibility(Professional developer tools for building any application type), Productivity(Powerful features such as CodeLens improve your team's productivity), Collaboration(Agile project planning tools, charts, etc.) and Subscriber benefits like Microsoft software, plus Azure, Pluralsight, etc.

Enterprise: It is an integrated, end to end solution for teams of any size with the demanding quality and scale needs. Microsoft provides a 90-days free trial of this edition and after the trial period, the user has to pay to continue using it. The main benefit of this edition is that it is highly scalable and deliver high-quality software.

Getting Started with Visual Studio 2017

• First, you must download and install the Visual Studio. For that, you can refer to Downloading and Installing Visual Studio 2017. Do not forget to select the .NET core workload during the installation of VS 2017. If you forget then you must modify the installation.

• You can see several tool windows when you will open the Visual Studio and start writing your first program as follows:
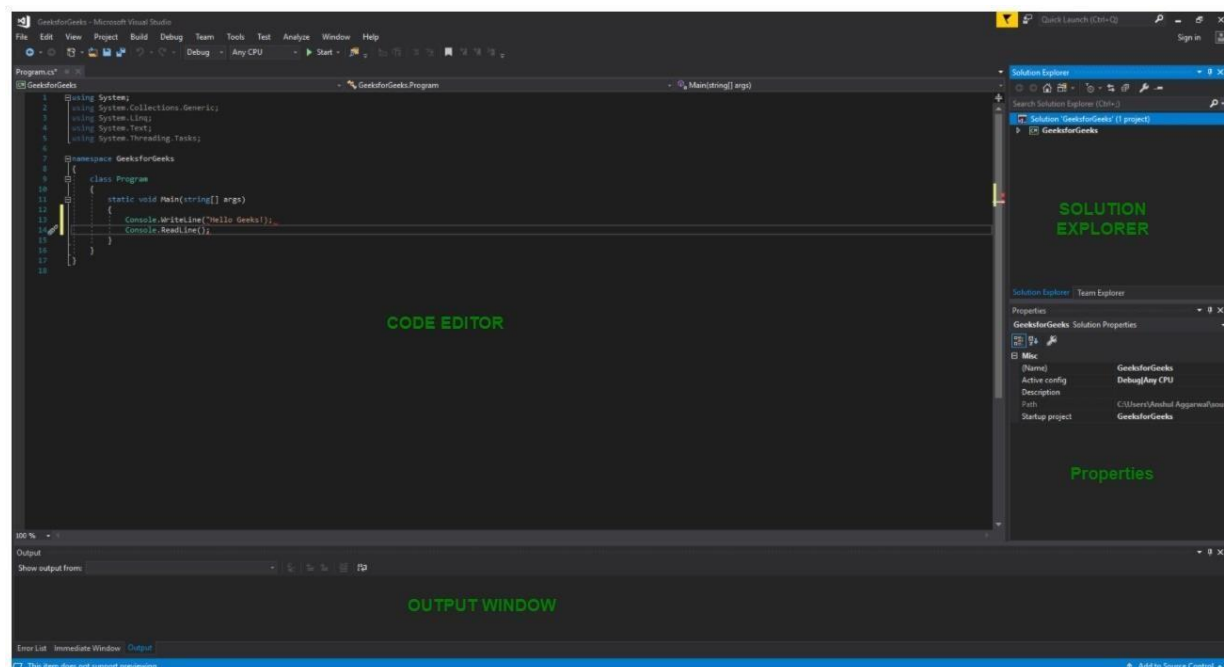
FIG 3(Code Editor)

1.   **Code Editor:** Where the user will write code.

2.   **Output Window**: Here the Visual Studio shows the outputs, compiler warnings, error messages and debugging information.

3.   **Solution Explorer:** It shows the files on which the user is currently working.

4.   **Properties:** It will give additional information and context about the selected parts of the current project.
•   A user can also add windows as per requirement by choosing them from View menu. In Visual Studio the tool windows are customizable as a user can add more windows, remove the existing open one or can move windows around to best suit.

15

• **Various Menus in Visual Studio:** A user can find a lot of menus on the top screen of Visual Studio as shown below
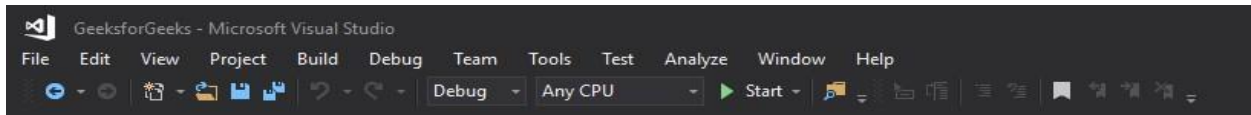
FIG 4(Menus)

1. Create, Open and save projects commands are contained by File menu.
2. Searching, Modifying, Refactoring code commands are contained by the Edit menu.
3. View Menu is used to open the additional tool windows in Visual Studio.

4. Project menu is used to add some files and dependencies in the project.
5. To change the settings, add functionality to Visual Studio via extensions, and access various Visual Studio tools can be used by using Tools menu.

The below menu is known as the toolbar which provide the quick access to the most frequently used commands. You can add and remove the commands by going to View → Customize
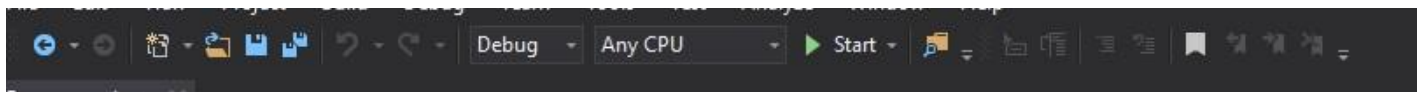
FIG 5(Menus)

Note:

• Support for different programming languages in Visual Studio is added by using a special VSPackage which is known as Language Service.

• When you will install the Visual Studio then the functionality which is coded as VSPackage will be available as Service.

• Visual Studio IDE provides the three different types of services known as SVsSolution, SVsUIShell, and SVsShell.

• SVsSolution service is used to provide the functionality to enumerate solutions and projects in Visual Studio.

• SVsUIShell service is used to provide User Interface functionality like toolbars, tabs etc.

• SvsShell service is used to deal with the registration of VSPackages.
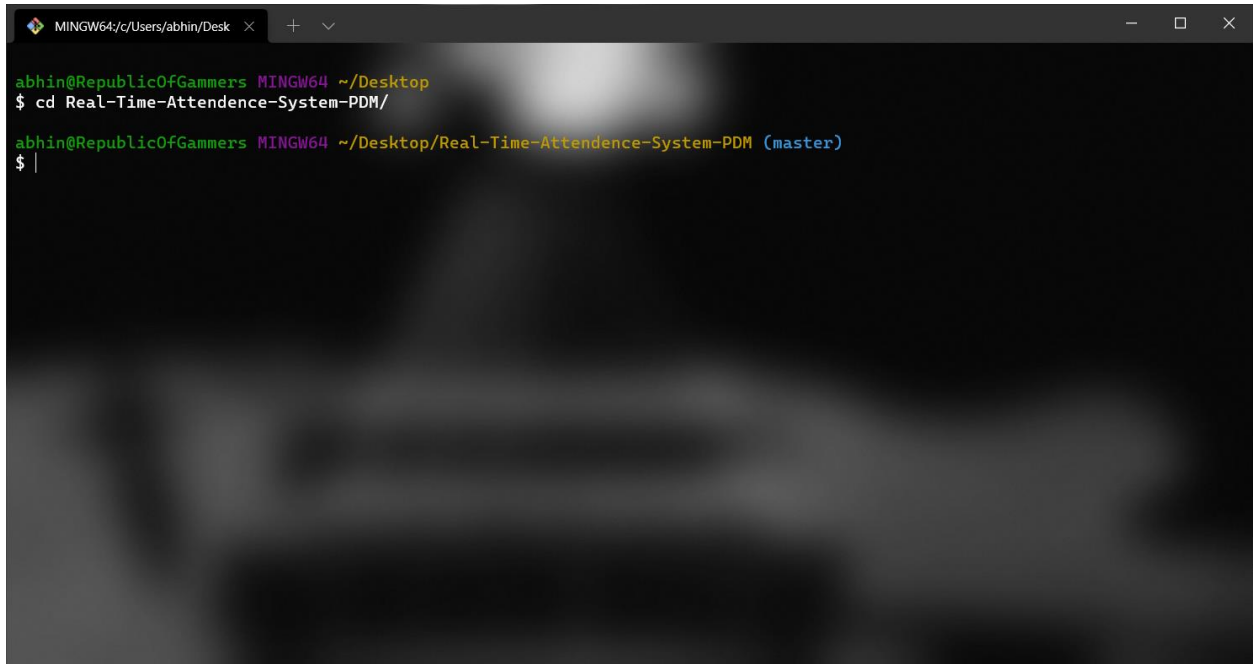
2.16 <u>GITBASH/GIT</u>

- Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.
- Git relies on the basis of distributed development of a software where more than one developer may have access to the source code of a specific application and can modify changes to it which may be seen by other developers..
- Initially designed and developed by Linus Torvalds for Linux kernel development in 2005.
- Every git working directory is a full-fledged repository with complete history and full version-tracking capabilities, independent of network access or a central server.
- Git allows a team of people to work together, all using the same files. And it helps the team cope up with the confusion that tends to happen when multiple people are editing the same files.

FIG 6 (GIT installation)

Git Bash is an application that provides Git command line experience on the Operating System. It is a command-line shell for enabling git with the command line in the system. A shell is a terminal application used to interface with an operating system through written commands. Git Bash is a package that installs Bash, some common bash utilities, and Git on a Windows operating system. In Git Bash the user interacts with the repository and git elements through the commands.

17

FIG 7 (GIT BASH)

2.17 <u>PYTHON 3.6.7</u>

Python is a high-level, general-purpose and a very popular programming language. Python programming language (latest Python 3) is being used in web development, Machine Learning applications, along with all cutting edge technology in Software Industry. Python Programming Language is very well suited for Beginners, also for experienced programmers with other programming languages like C++ and Java.

This specially designed Python tutorial will help you learn Python Programming Language in most efficient way, with the topics from basics to advanced (like Web-scraping, Django, Deep-Learning, etc.) with examples.

Below are some facts about Python Programming Language:

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms.

Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following:

- Machine Learning

- GUI Applications (like Kivy, Tkinter, PyQt etc.)

- Web frameworks like Django (used by YouTube, Instagram, Dropbox)

- Image processing (like OpenCV, Pillow)

- Web scraping (like Scrapy, BeautifulSoup, Selenium)

- Test frameworks

- Multimedia

- Scientific computing

- Text processing and many more..

FIG 8 (Python Installation)
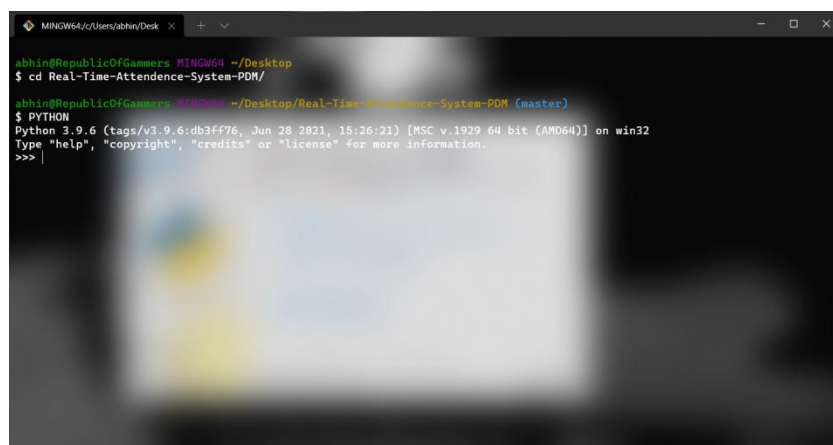
FIG 9 (Python shell)

CHAPTER 3

CODING

3.1 TOTAL FILES IN THE PROJECT

There are 5 folders and 2 files in the project.

Folders:-

1. Attendance
2. Image
3. StudentDetails
4. TrainingImage
5. TrainingImageLabel

Files :-

1. haarcascade_frontalface_default.xml
2. main.py
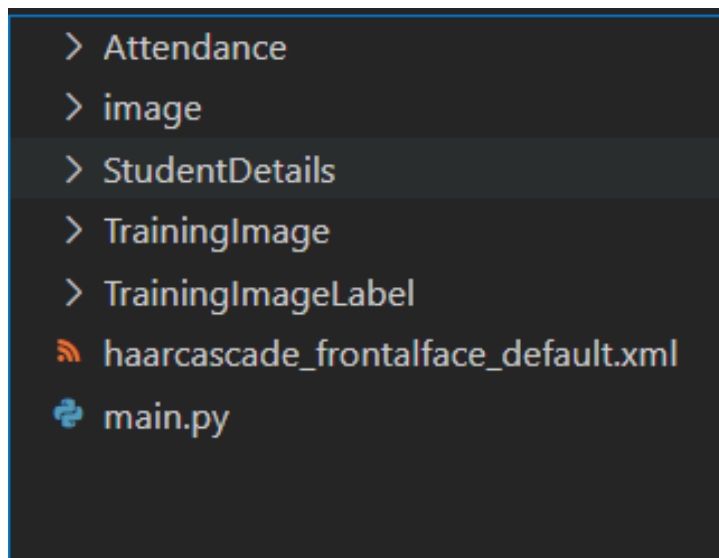
FIG 10(Files)

3.2 **Attendance folder** :- it contains all the attendance csv files that out machine learning model is going to generated.
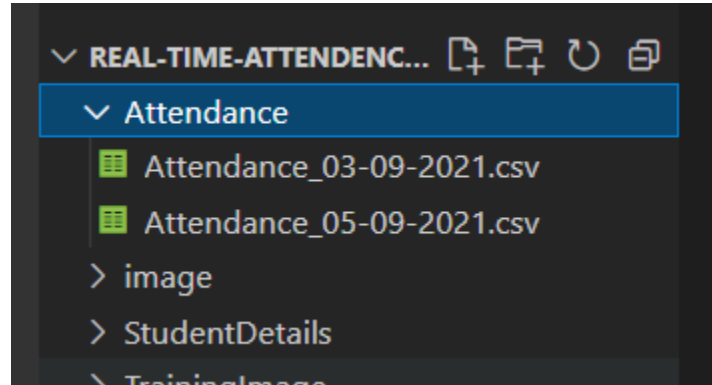
FIG 11(Attendance File)
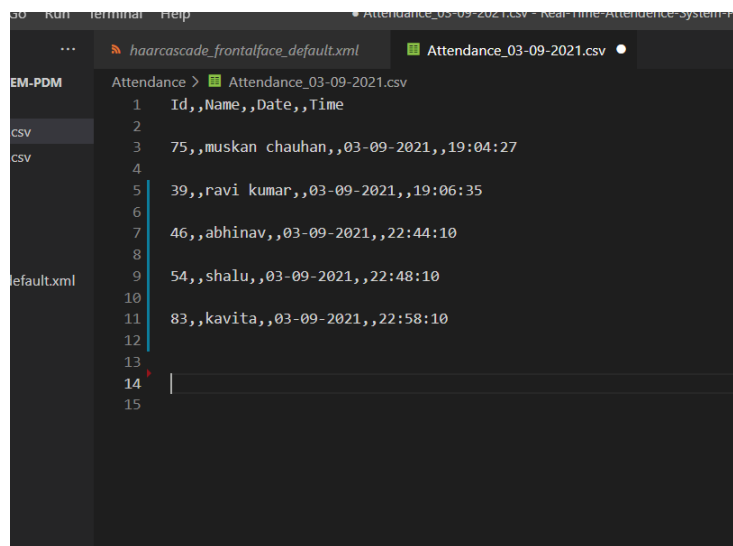
FIG 12(Attendance.csv)

3.3 **Image** :- it contains our project asset ie. Images of our projects
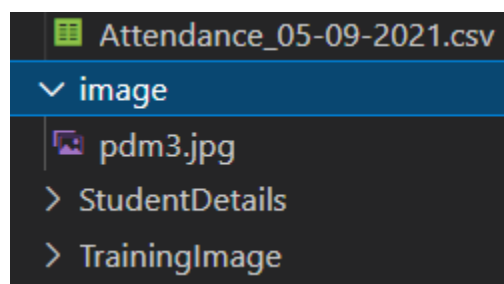
FIG 13(Image file)

3.4 **StudentsDetails** :- it contain the data of register student.
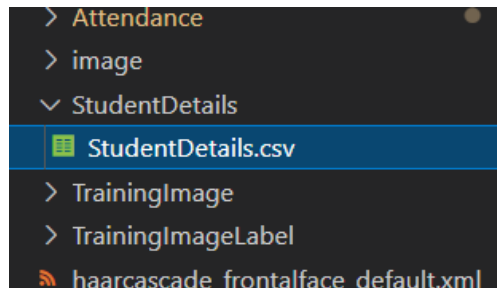
FIG 14(StudentsDetails)
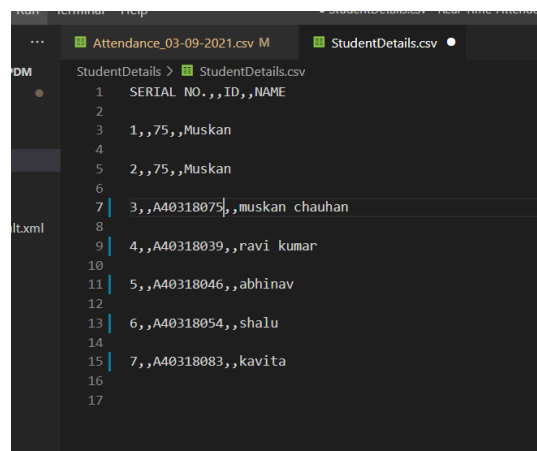
FIG 15 (StudentsDetails.csv)

3.5 **TrainingImage** :- it contains all the images that our program takes and help the method to train the model for face detection and object detechtion

FIG 16 (Training Image)

22

3.6 **TrainingLables** :- Here all the works happens it contains data of the student merged with the image. It trains the model to recognise the image of the particular student. It have a tranner.yml file which contains the image data in RGB formate in mattrices.

FIG 17 (Training.yml)

3.7 **Haarcascade_frontalface_default.xml** :- this is the largest dataset for image classification it contains a huge tone of images in RGB mattrices. Due to this data set we have to train our machine learning model in very less GPU requirements.

FIG 18 (Haarcascade.xml)

3.8 **Main.py** :- This file contains our ml model with open cv and gui interface code .

```
#############################################IMPORTING
################################################
import tkinter as tk
from tkinter import *
from tkinter import ttk
from tkinter import messagebox as mess
import tkinter.simpledialog as tsd
import cv2,os
import csv
import numpy as np
from PIL import Image
import pandas as pd
import datetime
import time
from PIL import ImageTk,Image


#############################################FUNCTIONS
#################################################


def assure_path_exists(path):
    dir = os.path.dirname(path)
    if not os.path.exists(dir):
        os.makedirs(dir)


##############################################################################
####
```

```python
def tick():
    time_string = time.strftime('%H:%M:%S')
    clock.config(text=time_string)
    clock.after(200,tick)


###############################################################################
#####

def contact():
    mess._show(title='Contact        us',        message="Please        contact        us        on        :
'chauhanmuskan2001@gmail.com' ")


###############################################################################
#####

def check_haarcascadefile():
    exists = os.path.isfile("haarcascade_frontalface_default.xml")
    if exists:
        pass
    else:
        mess._show(title='Some file missing', message='Please contact us for help')
        window.destroy()


###############################################################################
#####

def save_pass():
    assure_path_exists("TrainingImageLabel/")
    exists1 = os.path.isfile("TrainingImageLabel\psd.txt")
    if exists1:
```

```python
        tf = open("TrainingImageLabel\psd.txt", "r")
        key = tf.read()
    else:
        master.destroy()
        new_pas = tsd.askstring('Old Password not found', 'Please enter a new password below',
show='*')
        if new_pas == None:
            mess._show(title='No Password Entered', message='Password not set!! Please try again')
        else:
            tf = open("TrainingImageLabel\psd.txt", "w")
            tf.write(new_pas)
            mess._show(title='Password  Registered',  message='New  password  was  registered
successfully!!')
        return
    op = (old.get())
    newp= (new.get())
    nnewp = (nnew.get())
    if (op == key):
        if(newp == nnewp):
            txf = open("TrainingImageLabel\psd.txt", "w")
            txf.write(newp)
        else:
            mess._show(title='Error', message='Confirm new password again!!!')
            return
    else:
        mess._show(title='Wrong Password', message='Please enter correct old password.')
        return
    mess._show(title='Password Changed', message='Password changed successfully!!')
    master.destroy()
```

```python
###########################################################################
#####

def change_pass():
    global master
    master = tk.Tk()
    master.geometry("400x160")
    master.resizable(False,False)
    master.title("Change Password")
    master.configure(background="white")
    lbl4 = tk.Label(master,text='    Enter Old Password',bg='white',font=('times', 12, ' bold '))
    lbl4.place(x=10,y=10)
    global old
    old=tk.Entry(master,width=25 ,fg="black",relief='solid',font=('times', 12, ' bold '),show='*')
    old.place(x=180,y=10)
    lbl5 = tk.Label(master, text='   Enter New Password', bg='white', font=('times', 12, ' bold '))
    lbl5.place(x=10, y=45)
    global new
    new = tk.Entry(master, width=25, fg="black",relief='solid', font=('times', 12, ' bold '),show='*')
    new.place(x=180, y=45)
    lbl6 = tk.Label(master, text='Confirm New Password', bg='white', font=('times', 12, ' bold '))
    lbl6.place(x=10, y=80)
    global nnew
    nnew = tk.Entry(master, width=25, fg="black", relief='solid',font=('times', 12, ' bold '),show='*')
    nnew.place(x=180, y=80)
    cancel=tk.Button(master,text="Cancel", command=master.destroy ,fg="black" ,bg="red" ,height=1,width=25 , activebackground = "white" ,font=('times', 10, ' bold '))
    cancel.place(x=200, y=120)
```

```python
    save1 = tk.Button(master, text="Save", command=save_pass, fg="black", bg="#3ece48",
height = 1,width=25, activebackground="white", font=('times', 10, ' bold '))
    save1.place(x=10, y=120)
    master.mainloop()
```

###
#######

```python
def psw():
    assure_path_exists("TrainingImageLabel/")
    exists1 = os.path.isfile("TrainingImageLabel\psd.txt")
    if exists1:
        tf = open("TrainingImageLabel\psd.txt", "r")
        key = tf.read()
    else:
        new_pas = tsd.askstring('Old Password not found', 'Please enter a new password below',
show='*')
        if new_pas == None:
            mess._show(title='No Password Entered', message='Password not set!! Please try again')
        else:
            tf = open("TrainingImageLabel\psd.txt", "w")
            tf.write(new_pas)
            mess._show(title='Password  Registered', message='New  password  was  registered
successfully!!')
        return
    password = tsd.askstring('Password', 'Enter Password', show='*')
    if (password == key):
        TrainImages()
    elif (password == None):
        pass
```

```python
    else:
        mess._show(title='Wrong Password', message='You have entered wrong password')


###############################################################################
########

def clear():
    txt.delete(0, 'end')
    res = "1)Take Images  >>>  2)Save Profile"
    message1.configure(text=res)



def clear2():
    txt2.delete(0, 'end')
    res = "1)Take Images  >>>  2)Save Profile"
    message1.configure(text=res)


###############################################################################
#########

def TakeImages():
    check_haarcascadefile()
    columns = ['SERIAL NO.', '', 'ID', '', 'NAME']
    assure_path_exists("StudentDetails/")
    assure_path_exists("TrainingImage/")
    serial = 0
    exists = os.path.isfile("StudentDetails\StudentDetails.csv")
    if exists:
        with open("StudentDetails\StudentDetails.csv", 'r') as csvFile1:
            reader1 = csv.reader(csvFile1)
```

```python
        for l in reader1:
            serial = serial + 1
        serial = (serial // 2)
        csvFile1.close()
    else:
        with open("StudentDetails\StudentDetails.csv", 'a+') as csvFile1:
            writer = csv.writer(csvFile1)
            writer.writerow(columns)
            serial = 1
        csvFile1.close()
    Id = (txt.get())
    name = (txt2.get())
    if ((name.isalpha()) or (' ' in name)):
        cam = cv2.VideoCapture(0)
        harcascadePath = "haarcascade_frontalface_default.xml"
        detector = cv2.CascadeClassifier(harcascadePath)
        sampleNum = 0
        while (True):
            ret, img = cam.read()
            gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            faces = detector.detectMultiScale(gray, 1.3, 5)
            for (x, y, w, h) in faces:
                cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
                # incrementing sample number
                sampleNum = sampleNum + 1
                # saving the captured face in the dataset folder TrainingImage
                cv2.imwrite("TrainingImage\ " + name + "." + str(serial) + "." + Id + '.' + str(sampleNum)
+ ".jpg",
                            gray[y:y + h, x:x + w])
                # display the frame
```

```python
            cv2.imshow('Taking Images', img)
        # wait for 100 miliseconds
        if cv2.waitKey(100) & 0xFF == ord('q'):
            break
        # break if the sample number is morethan 100
        elif sampleNum > 100:
            break
    cam.release()
    cv2.destroyAllWindows()
    res = "Images Taken for ID : " + Id
    row = [serial, '', Id, '', name]
    with open('StudentDetails\StudentDetails.csv', 'a+') as csvFile:
        writer = csv.writer(csvFile)
        writer.writerow(row)
    csvFile.close()
    message1.configure(text=res)
else:
    if (name.isalpha() == False):
        res = "Enter Correct name"
        message.configure(text=res)


##############################################################################
##########

def TrainImages():
    check_haarcascadefile()
    assure_path_exists("TrainingImageLabel/")
    recognizer = cv2.face_LBPHFaceRecognizer.create()
    harcascadePath = "haarcascade_frontalface_default.xml"
    detector = cv2.CascadeClassifier(harcascadePath)
```

```python
    faces, ID = getImagesAndLabels("TrainingImage")
    try:
        recognizer.train(faces, np.array(ID))
    except:
        mess._show(title='No Registrations', message='Please Register someone first!!!')
        return
    recognizer.save("TrainingImageLabel\Trainner.yml")
    res = "Profile Saved Successfully"
    message1.configure(text=res)
    message.configure(text='Total Registrations till now  : ' + str(ID[0]))


#############################################################################
##############

def getImagesAndLabels(path):
    # get the path of all the files in the folder
    imagePaths = [os.path.join(path, f) for f in os.listdir(path)]
    # create empth face list
    faces = []
    # create empty ID list
    Ids = []
    # now looping through all the image paths and loading the Ids and the images
    for imagePath in imagePaths:
        # loading the image and converting it to gray scale
        pilImage = Image.open(imagePath).convert('L')
        # Now we are converting the PIL image into numpy array
        imageNp = np.array(pilImage, 'uint8')
        # getting the Id from the image
        ID = int(os.path.split(imagePath)[-1].split(".")[1])
        # extract the face from the training image sample
```

```python
        faces.append(imageNp)

        Ids.append(ID)

    return faces, Ids


############################################################################
#############

def TrackImages():
    check_haarcascadefile()
    assure_path_exists("Attendance/")
    assure_path_exists("StudentDetails/")
    for k in tv.get_children():
        tv.delete(k)
    msg = ''
    i = 0
    j = 0
    recognizer = cv2.face.LBPHFaceRecognizer_create()  # cv2.createLBPHFaceRecognizer()
    exists3 = os.path.isfile("TrainingImageLabel\Trainner.yml")
    if exists3:
        recognizer.read("TrainingImageLabel\Trainner.yml")
    else:
        mess._show(title='Data Missing', message='Please click on Save Profile to reset data!!')
        return
    harcascadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(harcascadePath);

    cam = cv2.VideoCapture(0)
    font = cv2.FONT_HERSHEY_SIMPLEX
    col_names = ['Id', '', 'Name', '', 'Date', '', 'Time']
    exists1 = os.path.isfile("StudentDetails\StudentDetails.csv")
```

```python
if exists1:
    df = pd.read_csv("StudentDetails\StudentDetails.csv")
else:
    mess._show(title='Details Missing', message='Students details are missing, please check!')
    cam.release()
    cv2.destroyAllWindows()
    window.destroy()
while True:
    ret, im = cam.read()
    gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(gray, 1.2, 5)
    for (x, y, w, h) in faces:
        cv2.rectangle(im, (x, y), (x + w, y + h), (225, 0, 0), 2)
        serial, conf = recognizer.predict(gray[y:y + h, x:x + w])
        if (conf < 50):
            ts = time.time()
            date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')
            timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
            aa = df.loc[df['SERIAL NO.'] == serial]['NAME'].values
            ID = df.loc[df['SERIAL NO.'] == serial]['ID'].values
            ID = str(ID)
            ID = ID[1:-1]
            bb = str(aa)
            bb = bb[2:-2]
            attendance = [str(ID), '', bb, '', str(date), '', str(timeStamp)]

        else:
            Id = 'Unknown'
            bb = str(Id)
        cv2.putText(im, str(bb), (x, y + h), font, 1, (255, 255, 255), 2)
```

```python
        cv2.imshow('Taking Attendance', im)

        if (cv2.waitKey(1) == ord('q')):

            break

ts = time.time()

date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')

exists = os.path.isfile("Attendance\Attendance_" + date + ".csv")

if exists:

    with open("Attendance\Attendance_" + date + ".csv", 'a+') as csvFile1:

        writer = csv.writer(csvFile1)

        writer.writerow(attendance)

    csvFile1.close()

else:

    with open("Attendance\Attendance_" + date + ".csv", 'a+') as csvFile1:

        writer = csv.writer(csvFile1)

        writer.writerow(col_names)

        writer.writerow(attendance)

    csvFile1.close()

with open("Attendance\Attendance_" + date + ".csv", 'r') as csvFile1:

    reader1 = csv.reader(csvFile1)

    for lines in reader1:

        i = i + 1

        if (i > 1):

            if (i % 2 != 0):

                iidd = str(lines[0]) + '   '

                tv.insert('', 0, text=iidd, values=(str(lines[2]), str(lines[4]), str(lines[6])))

csvFile1.close()

cam.release()

cv2.destroyAllWindows()
```

```python
#######################################USED                    STUFFS
#############################################

global key
key = ''

ts = time.time()
date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')
day,month,year=date.split("-")

mont={'01':'January',
    '02':'February',
    '03':'March',
    '04':'April',
    '05':'May',
    '06':'June',
    '07':'July',
    '08':'August',
    '09':'September',
    '10':'October',
    '11':'November',
    '12':'December'
    }
#######################################GUI                    FRONT-END
#############################################

window = tk.Tk()
window.geometry("1280x720")
window.resizable(True,False)
window.title("PDM Attendance System")
```

```python
########## background image ##########
img = ImageTk.PhotoImage(Image.open("image/pdm3.jpg"))
panel = tk.Label(window, image=img)
panel.pack(fill="both",expand=True)
panel.configure(image=img)
panel.image = img

window.configure(background='#0c0c0d') #black color

frame1 = tk.Frame(window, bg="#f06b2e") #black
frame1.place(relx=0.11, rely=0.17, relwidth=0.39, relheight=0.80)

frame2 = tk.Frame(window, bg="#f06b2e")
frame2.place(relx=0.51, rely=0.17, relwidth=0.38, relheight=0.80)

message3 = tk.Label(window, text="PDM Face Recognition Based Attendance System"
,fg="Black" ,width=55 ,height=1,font=('times', 29, ' bold '))
message3.place(x=10, y=10)

frame3 = tk.Frame(window, bg="#c4c6ce") #light grey color
frame3.place(relx=0.52, rely=0.09, relwidth=0.09, relheight=0.07)

frame4 = tk.Frame(window, bg="#c4c6ce")
frame4.place(relx=0.36, rely=0.09, relwidth=0.16, relheight=0.07)

datef = tk.Label(frame4, text = day+"-"+mont[month]+"-"+year+"  |  ", fg="black",bg="#FFFFFF"
,width=55 ,height=1,font=('times', 22, ' bold '))
datef.pack(fill='both',expand=1)

clock = tk.Label(frame3,fg="black",bg="#FFFFFF" ,width=55 ,height=1,font=('times', 22, ' bold '))
```

```python
clock.pack(fill='both',expand=1)
tick()


head2 = tk.Label(frame2, text="          For  New  Registrations                    ",
fg="black",bg="#ffffff" ,font=('times', 17, ' bold ') ) #fffff is white color
head2.grid(row=0,column=0)


head1 = tk.Label(frame1, text="              For  Already  Registered                ",
fg="black",bg="#ffffff" ,font=('times', 17, ' bold ') )
head1.place(x=0,y=0)


lbl = tk.Label(frame2, text="Enter  ID",width=20    ,height=1    ,fg="black"    ,bg="#00aeff"
,font=('times', 17, ' bold ') )
lbl.place(x=80, y=55)


txt = tk.Entry(frame2,width=32 ,fg="black",font=('times', 15, ' bold '))
txt.place(x=30, y=88)


lbl2 = tk.Label(frame2, text="Enter Name",width=20   ,fg="black"   ,bg="#00aeff" ,font=('times',
17, ' bold ')) #00aeff blue color
lbl2.place(x=80, y=140)


txt2 = tk.Entry(frame2,width=32 ,fg="black",font=('times', 15, ' bold ')  )
txt2.place(x=30, y=173)


message1 = tk.Label(frame2, text="1)Take  Images    >>>    2)Save  Profile" ,bg="#00aeff"
,fg="black"  ,width=39 ,height=1, activebackground = "yellow" ,font=('times', 15, ' bold '))
message1.place(x=7, y=230)
```

```
message   =   tk.Label(frame2,   text=""   ,bg="#00aeff"   ,fg="black"        ,width=39,height=1,
activebackground = "yellow" ,font=('times', 16, ' bold '))
message.place(x=7, y=450)


lbl3  =  tk.Label(frame1,  text="Attendance",width=20    ,fg="black"   ,bg="#00aeff"   ,height=1
,font=('times', 17, ' bold '))
lbl3.place(x=100, y=115)


res=0
exists = os.path.isfile("StudentDetails\StudentDetails.csv")
if exists:
    with open("StudentDetails\StudentDetails.csv", 'r') as csvFile1:
        reader1 = csv.reader(csvFile1)
        for l in reader1:
            res = res + 1
    res = (res // 2) - 1
    csvFile1.close()
else:
    res = 0
message.configure(text='Total Registrations till now  : '+str(res))


##################### MENUBAR ###############################

menubar = tk.Menu(window,relief='ridge')
filemenu = tk.Menu(menubar,tearoff=0)
filemenu.add_command(label='Change Password', command = change_pass)
filemenu.add_command(label='Contact Us', command = contact)
filemenu.add_command(label='Exit',command = window.destroy)
menubar.add_cascade(label='Help',font=('times', 29, ' bold '),menu=filemenu)
```

```
################### TREEVIEW ATTENDANCE TABLE ###################

tv= ttk.Treeview(frame1,height =13,columns = ('name','date','time'))
tv.column('#0',width=82)
tv.column('name',width=130)
tv.column('date',width=133)
tv.column('time',width=133)
tv.grid(row=2,column=0,padx=(0,0),pady=(150,0),columnspan=4)
tv.heading('#0',text ='ID')
tv.heading('name',text ='NAME')
tv.heading('date',text ='DATE')
tv.heading('time',text ='TIME')


###################### SCROLLBAR ############################

scroll=ttk.Scrollbar(frame1,orient='vertical',command=tv.yview)
scroll.grid(row=2,column=4,padx=(0,100),pady=(150,0),sticky='ns')
tv.configure(yscrollcommand=scroll.set)


###################### BUTTONS ############################

clearButton = tk.Button(frame2, text="Clear", command=clear   ,fg="black"   ,bg="#ea2a2a"
,width=11 ,activebackground = "white" ,font=('times', 11, ' bold '))
clearButton.place(x=335, y=86)
clearButton2 = tk.Button(frame2, text="Clear", command=clear2   ,fg="black"   ,bg="#ea2a2a"
,width=11 , activebackground = "white" ,font=('times', 11, ' bold '))
clearButton2.place(x=335, y=172)
takeImg   =   tk.Button(frame2,   text="Take   Images",   command=TakeImages      ,fg="white"
,bg="blue"  ,width=34 ,height=1, activebackground = "white" ,font=('times', 15, ' bold '))
takeImg.place(x=30, y=300)
```

```python
trainImg = tk.Button(frame2, text="Save  Profile", command=psw ,fg="white"    ,bg="blue"
,width=34 ,height=1, activebackground = "white" ,font=('times', 15, ' bold '))
trainImg.place(x=30, y=380)
trackImg = tk.Button(frame1, text="Take  Attendance", command=TrackImages    ,fg="black"
,bg="yellow" ,width=35 ,height=1, activebackground = "white" ,font=('times', 15, ' bold '))
trackImg.place(x=30,y=50)
quitWindow  =  tk.Button(frame1,  text="Quit",  command=window.destroy      ,fg="black"
,bg="red"  ,width=35 ,height=1, activebackground = "white" ,font=('times', 15, ' bold '))
quitWindow.place(x=30, y=450)


##################### END ###################################

window.configure(menu=menubar)
window.mainloop()


###################################################################################
######################
```

CHAPTER 4

TESTING & RESULT

4.1 BASIC VIEW OF THE PROJECT
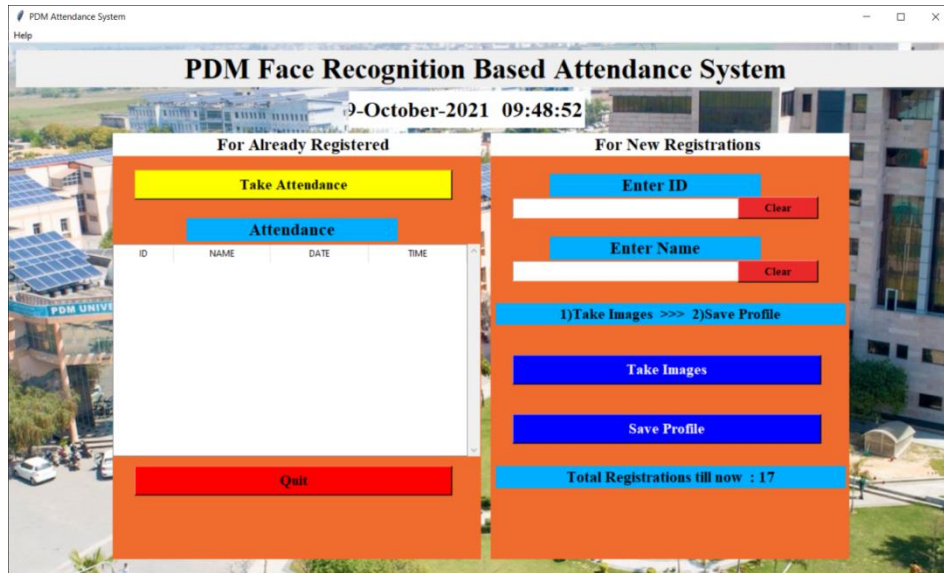
FIG 19(Basic view of the Project)

4.2 WHEN SOMEONE REGISTER YOURSELF
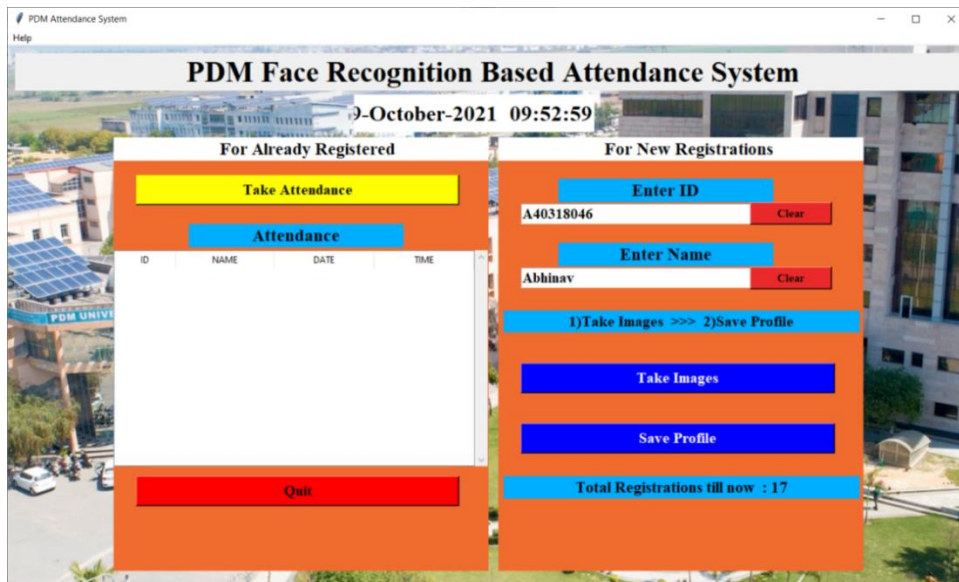
FIG 20 (Abhinav's Registration)
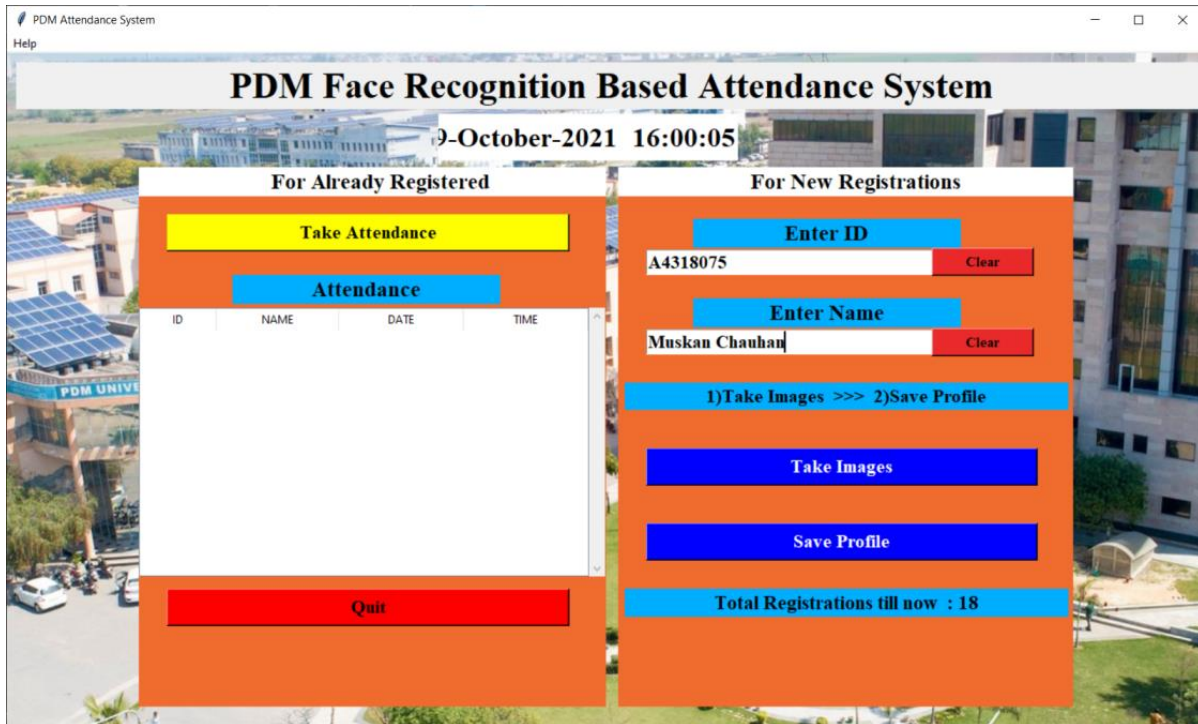
FIG 21(Muskan's Registration)
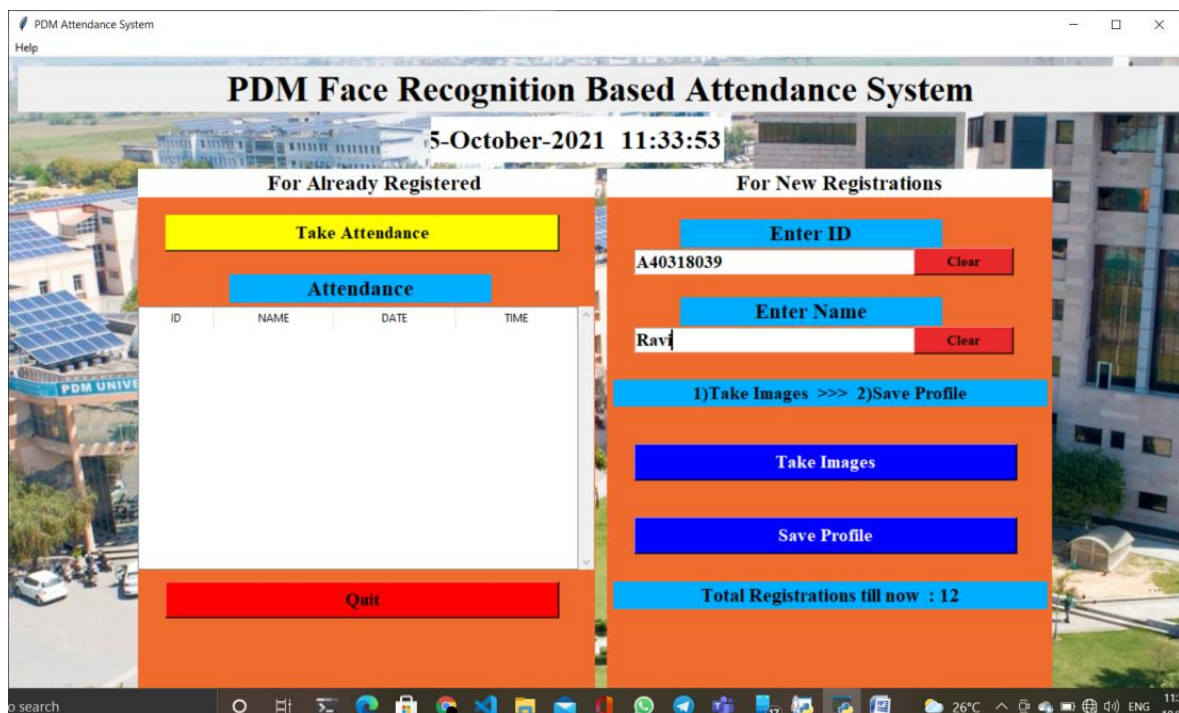
FIG 22 (Ravi's Registration)

FIG 23 (Shalu's Registration)

FIG 24 (Kavita's Registration)

4.3 FACE REGISTRATION

FIG 25 (Face Registration of Muskan)
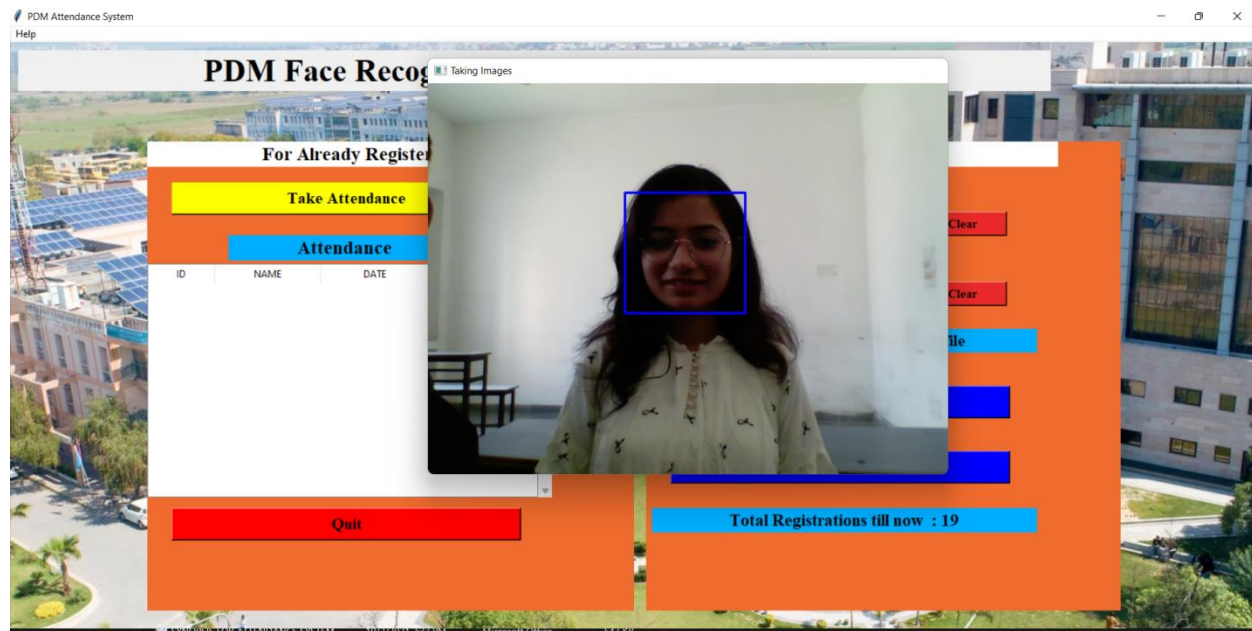
FIG 26(Face Registration of Shalu)
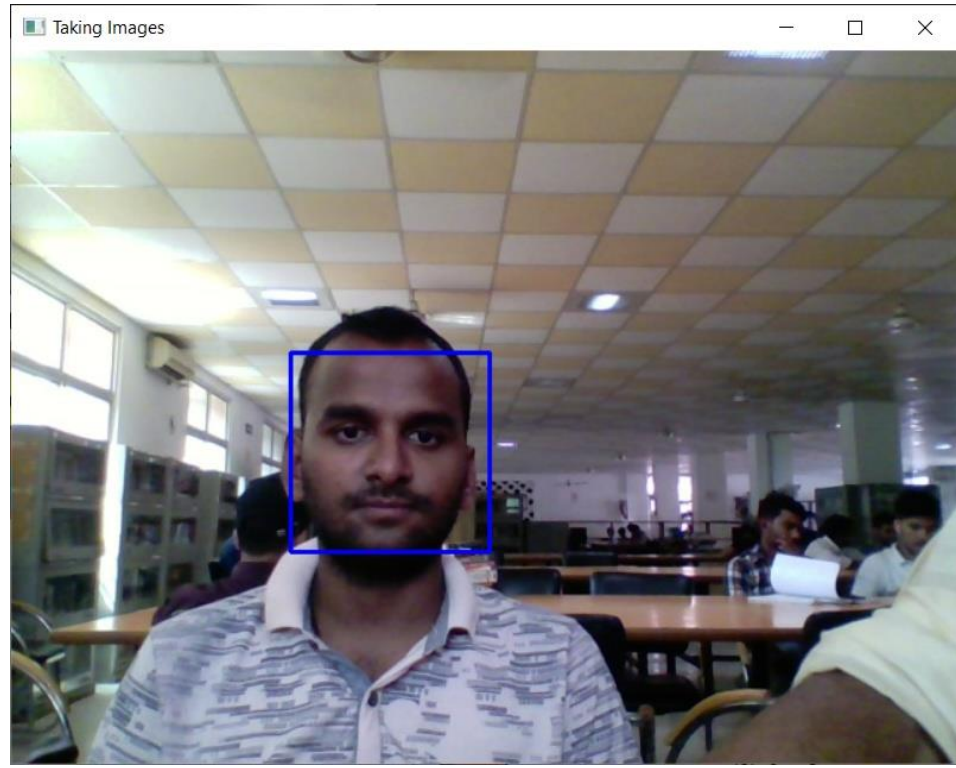
FIG 27(Face Registration of Ravi)
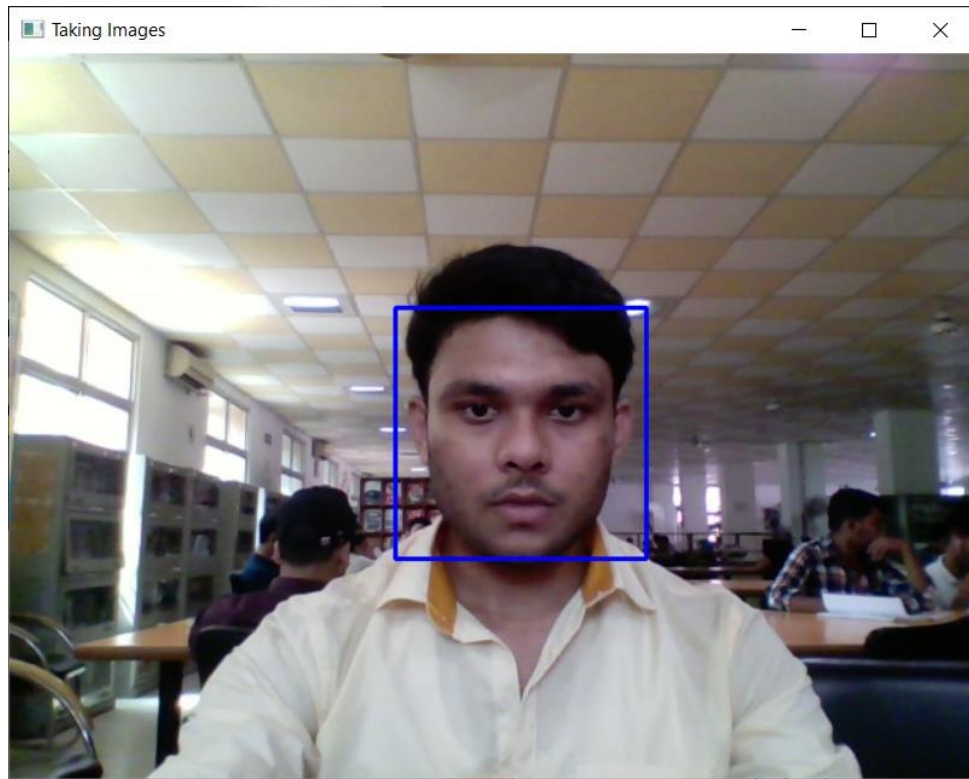
FIG 28 (Face Registration of Abhinav)
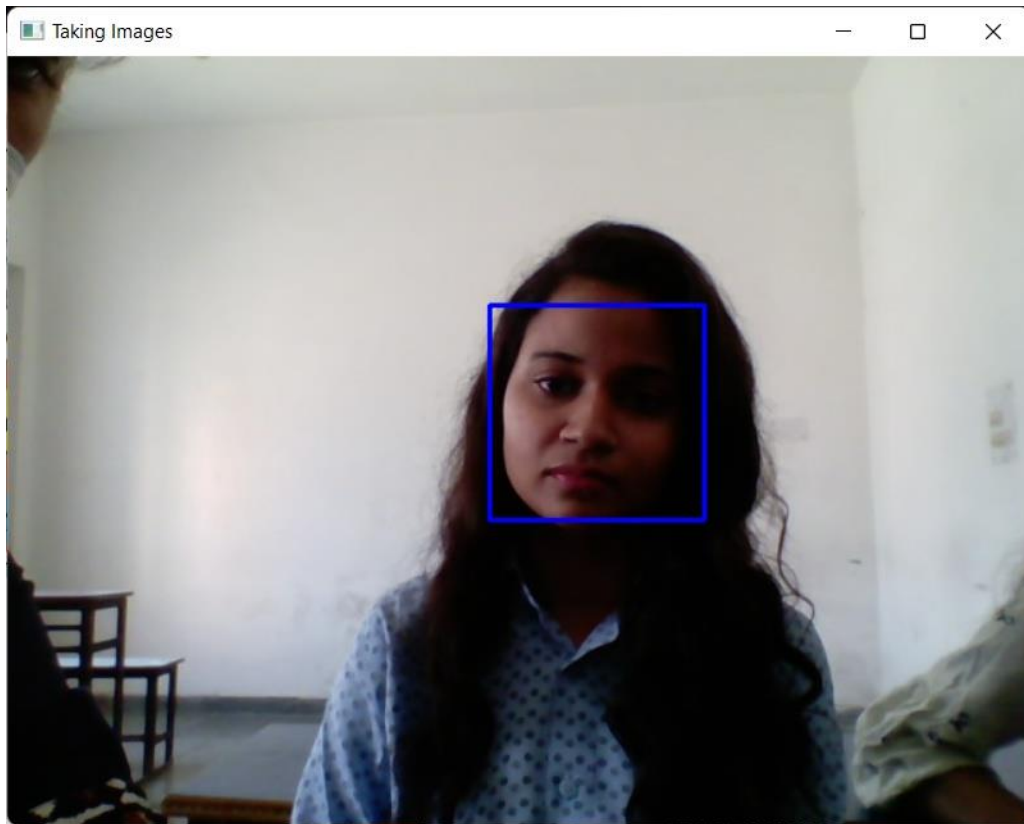
FIG 29 (Face Registration of Kavita)
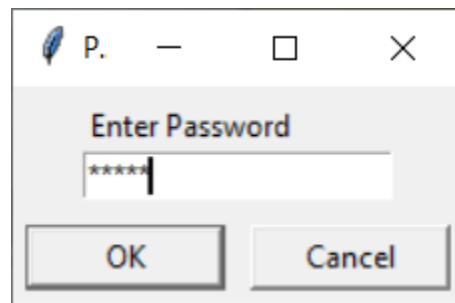
FIG 30 (Admin Password for Registration)

4.4 <u>ATTENDANCE BY FACE RECORGNIZATION :-</u>
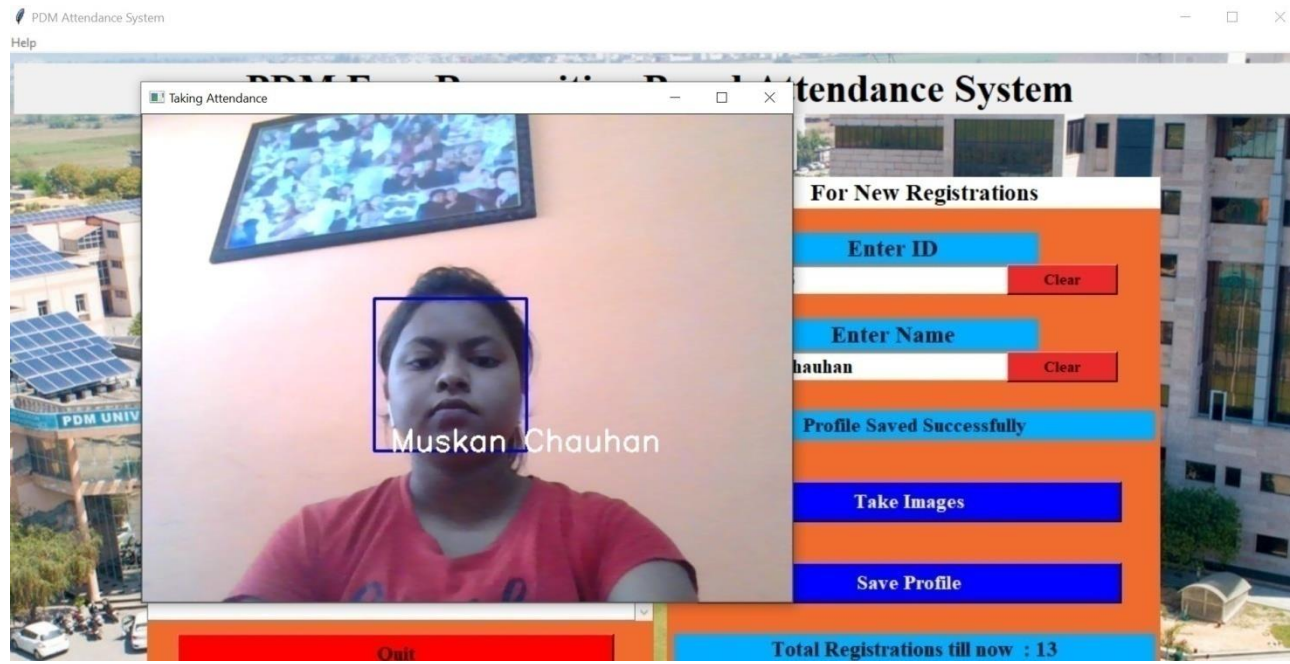
FIG 31(Attendance By Face Recognition of Muskan)
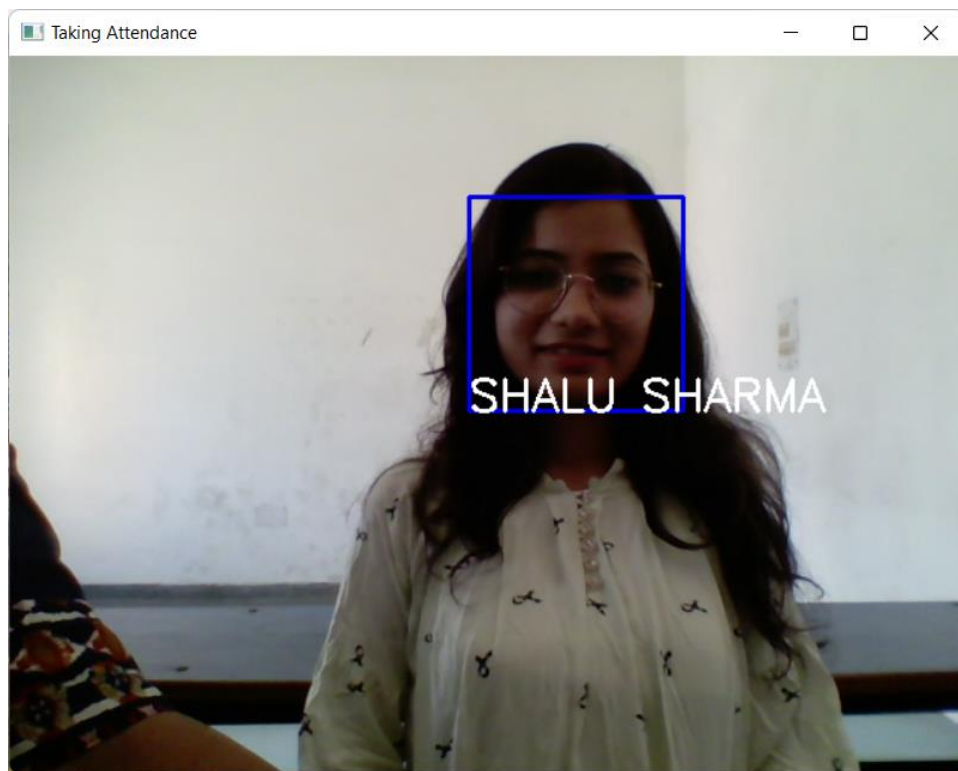
FIG 32(Attendance By Face Recognition of Shalu)
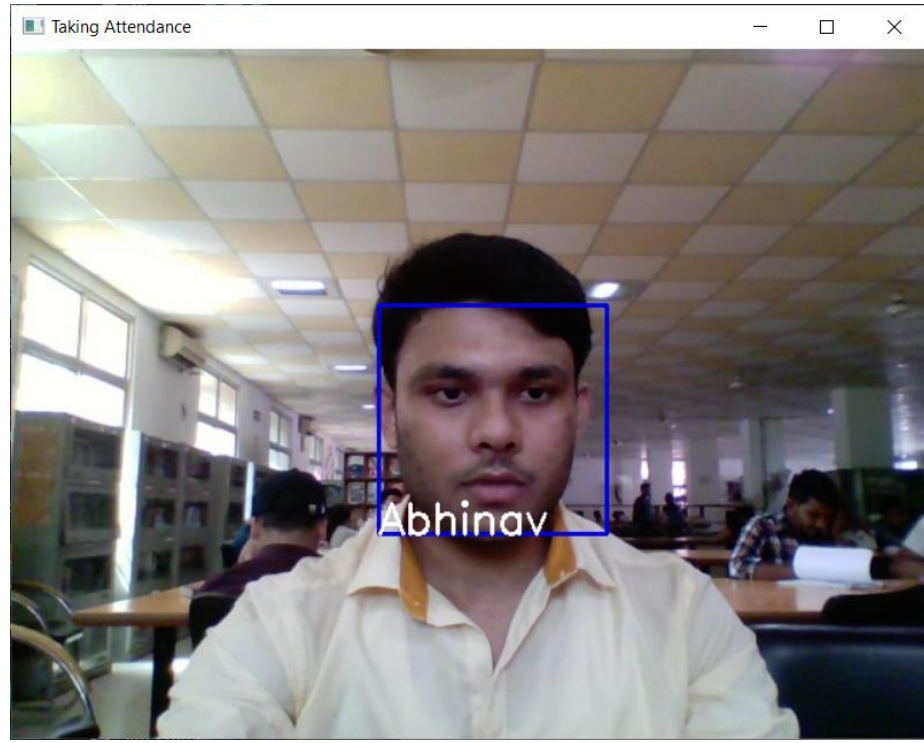
FIG 33 (A Attendance By Face Recognition of Abhinav)
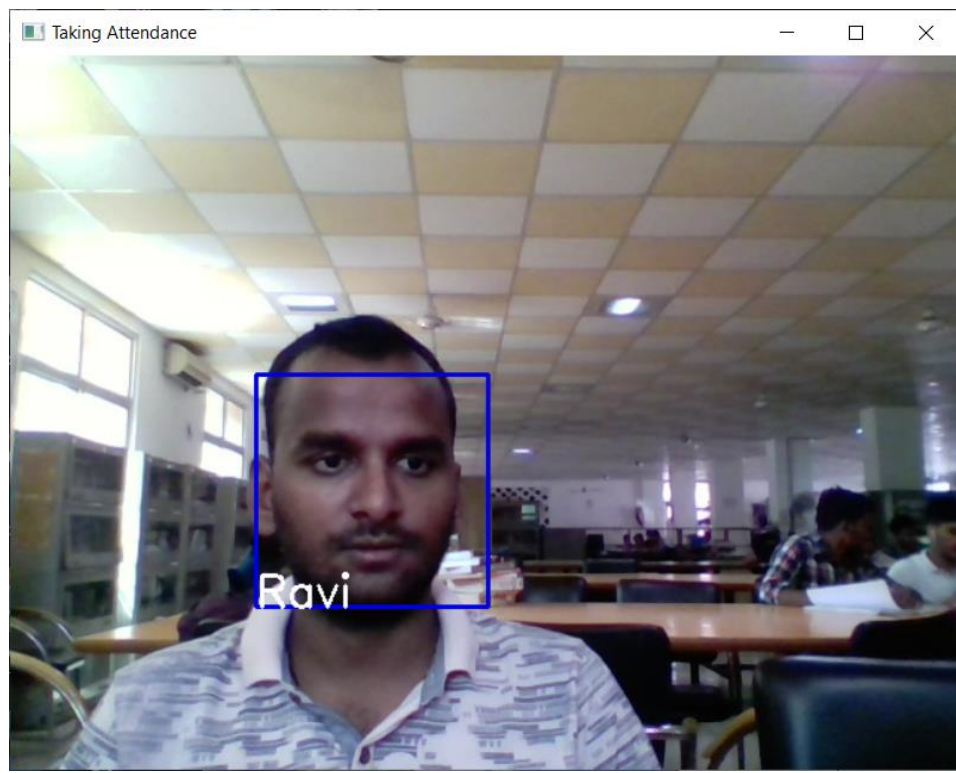
FIG 34 (Attendance By Face Recognition of Ravi)
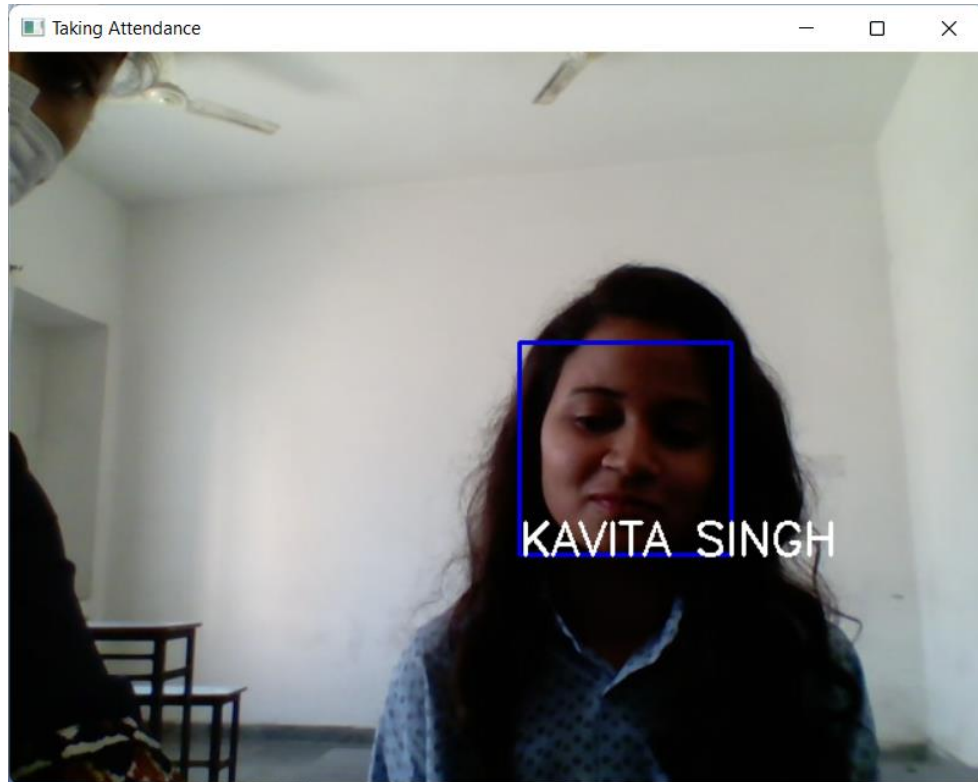
FIG 35 (Attendance By Face Recognition of Kavita)

4.5 <u>ATTENDANCE MARKED :-</u>

FIG 36 (Muskan's Attendance Marked)
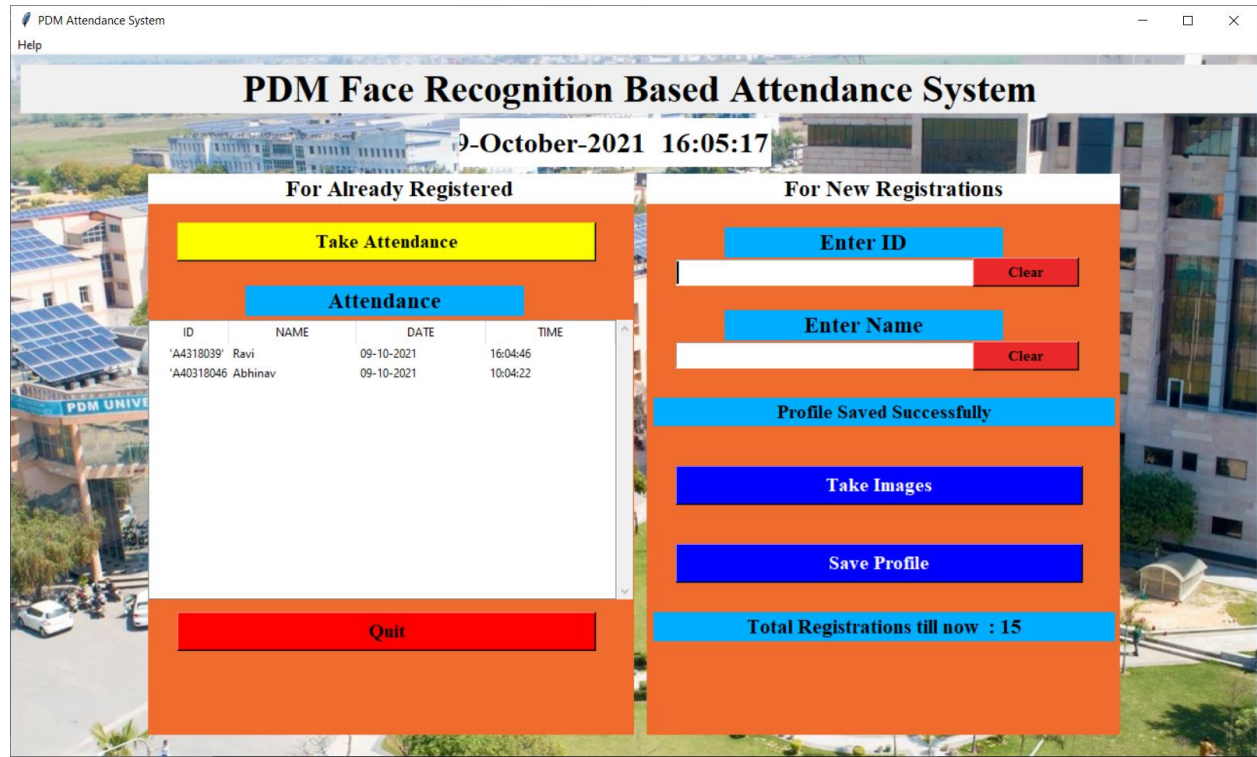
FIG 37 (Ravi and Abhinav's Attendance Marked)
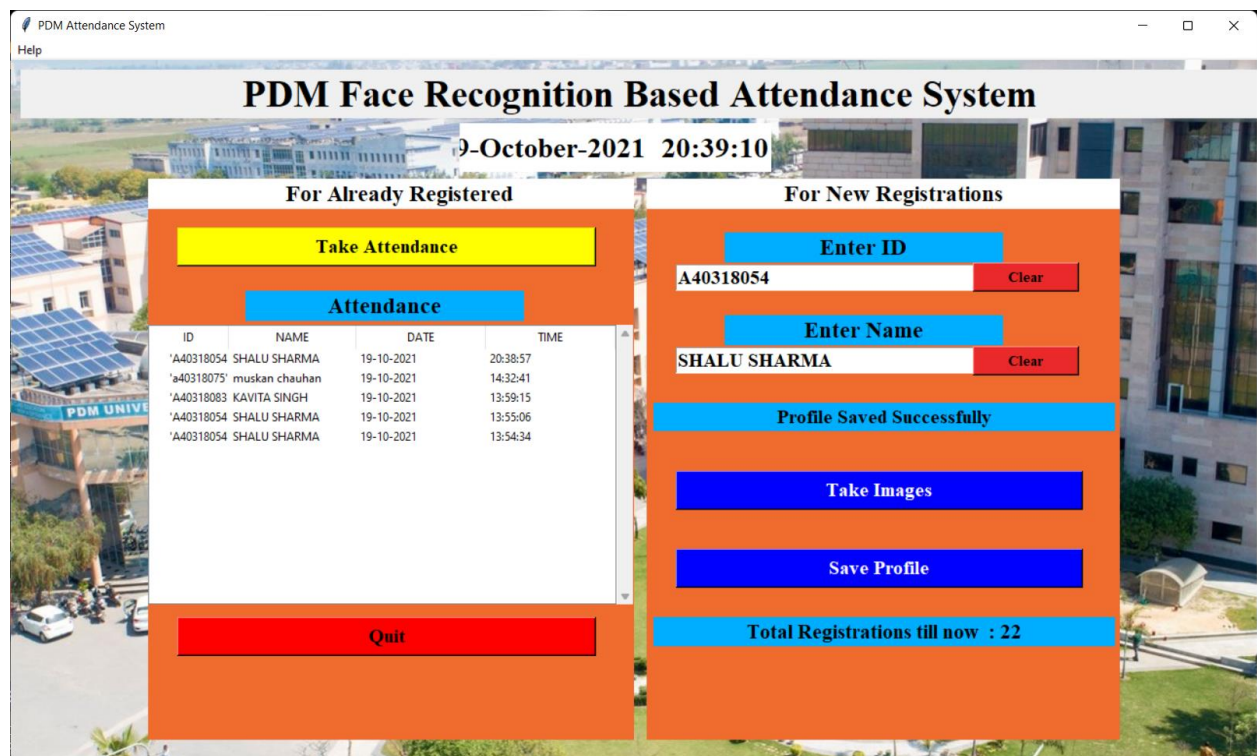
FIG 38 (Shalu and Kavita's Attendance Marked)

FIG 39 (Attendance.csv)

FIG 40 (Attendance.csv)

RESULT :-   we have successfully compiled our projects with some minor bug fixes such as background mirroring , chat message forwarding to different rooms and some textual bugs.

CHAPTER 5

CONCLUSION

No one is perfect, but humanity is inching closer to a more advanced civilization every day. Every technology that comes for the first time has many flaws and disadvantages. But all of this gets rectified slowly and steadily with time. Who would have thought that airplane for just a single person built by Wright brothers once would one day carry 1000s of tons of cargo at one go? These things take time to get perfect. Who knows what all we will be able to do with artificial intelligence in the future. This is all just a beginning. Let's see and enjoy this beautiful journey till it lasts.

At the end we have now successfully built Pdm face recognition attendance system and have a fair understanding that how open cv works, how to classify images, compression of image, render graphics and front-end and backend on PYTHON.

FUTURE SCOPE

A mail which contains the information about absent as well as attendance percentage is mailed to the respective parents instead of messages using face recognition based attendance management system. Process of taking attendance is done module by module, it is possible to update attendance in one click or touch instead. Parents can also get the information about the internal assessment marks through messages if their children.

BIBLIOGRAPHY

- https://www.python.org/
- https://github.com/opencv/opencv_contrib
- https://pypi.org/project/opencv-contrib-python/
- https://opencv.org/
- https://realpython.com/face-recognition-with-python/
- https://www.mygreatlearning.com/blog/face-recognition/
- https://docs.python.org/3/library/tkinter.html
- https://www.tutorialspoint.com/python/python_gui_programming.htm
- https://github.com/opencv/opencv/tree/master/data/haarcascades
- https://stackoverflow.com/questions/68436250/try-to-find-face-detection-using-opencv-harcascade-file-but-the-x-y-w-h-values-a
- https://stackoverflow.com/questions/14557814/attendance-tracker-using-excel-and-python
- https://numpy.org/doc/stable/user/whatisnumpy.html
- https://jivp-eurasipjournals.springeropen.com/articles/10.1186/s13640-019-0417-8
- https://github.com/getsanjeev/compression-DCT
- https://pillow.readthedocs.io/en/stable/
- https://docs.python.org/3/library/datetime.html
- https://pandas.pydata.org/
- https://docs.python.org/3/library/tkinter.html
- https://www.javatpoint.com/python-tkinter
- https://www.pythontutorial.net/tkinter/

REFERENCES

- https://www.python.org/
- https://docs.python.org/3/library/tkinter.html
- https://docs.python.org/3/library/os.html
- https://docs.python.org/3/library/datetime.html
- https://pypi.org/project/python-dateutil/
- https://numpy.org/
- https://docs.opencv.org/4.5.0/d6/d00/tutorial_py_root.html
- https://pillow.readthedocs.io/en/stable/
- https://pandas.pydata.org/
- https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b