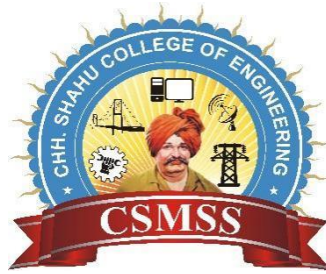


CSMSS
CHH. SHAHU COLLEGE OF ENGINEERING
KANCHANWADI, PAITHAN ROAD, AURANGABAD – 431011



A REPORT OF PROJECT PHASE-II

ON

Visitor Identification and Notification System

SUBMITTED IN PARTIAL FULFILLMENT OF REQUIREMENT FOR

AWARD OF THE DEGREE

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE & ENGINEERING

Submitted By

Mr Ruturaj Adhane (Roll No: CS4145)

Mr Saurabh Mehetre (Roll No: CS4143)

Mr Avishkar Gaikwad (Roll No: CS4151)

Under the guidance of

Dr. Vilas Gaikwad

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Academic Year: 2021-22Part-II
CSMSS

CHH. SHAHU COLLEGE OF ENGINEERING

KANCHANWADI, PAITHAN ROAD, AURANGABAD – 431011

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



CERTIFICATE

Certified that the Project (Phase-II) work entitled **“Visitor Identification and Notification System”** is bonafide work carried out by **Mr. Ruturaj Adhane**, Roll no: **CS4145**, **Mr. Saurabh Mehetre**, Roll no: **CS4143**, **Mr. Avishkar Gaikwad**, Roll no: **CS4151**, in partial fulfillment for award of the Degree **Bachelor of Technology** in **Computer Science & Engineering** of **Dr. Babasaheb Ambedkar Technological University**, Lonere, Raigad, during academic year 2021-22 Part-II.

Dr. Vilas Gaikwad

(Project Guide)

Dr. S. P. Abhang

(Head of the Department)

Dr. U. B. Shinde

(Principal)

DECLARATION

We, **Saurabh Sunil Mehetre, Ruturaj Dnyaneshwar Adhane, Avishkar Gaikwad & CS4143, CS4145, CS4151 of Project member's** students of 8th semester in B.Tech in Computer Science & Engineering, CSMSS Chh. Shahu College of Engineering, Aurangabad, hereby declare that the project work entitled “**Visitor Identification and Notification System**” submitted to the Dr. Babasaheb Ambedkar Technological University, Lonere, Raigad during academic year 2021-22, is a record of an original work done by us.

This project work is submitted in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science & Engineering.

The results embodied in this report have not been submitted to any University or Institute for the award of any degree.

Date: 14/07/2022

Place: Aurangabad

Avishkar Gaikwad (CS4151)

Saurabh Mehetre (CS4143)

Ruturaj Adhane (CS4145)

ACKNOWLEDGMENT

We take this opportunity to express our profound gratitude and deep regards to our project guide **Dr. Vilas Gaikwad** for his exemplary guidance, monitoring and constant encouragement throughout the course of this report.

We are thankful to **Dr. Vilas S. Gaikwad** Project Coordinator, **Dr. S. P. Abhang**, Head – Computer Science & Engineering department and **Dr. U. B. Shinde**, Principal- CSMSS Chh. Shahu College of Engineering, Aurangabad for their continuous support and encouragement throughout the academic.

We are also obliged to faculty members of Computer Science & Engineering Department, CSMSS Chh. Shahu College of Engineering, Aurangabad, for their valuable guidance and information provided by them in their respective fields.

Date: 14/07/2022

Place: Aurangabad

Avishkar Gaikwad (CS4151)

Saurabh Mehetre (CS4143)

Ruturaj Adhane (CS4145)

CONTENTS

1. Project Title
2. Abstract
3. Objective, Introduction &
Literature survey
4. Problem Statement
5. Methodology
6. Testing
7. Results and Analysis
8. Project Artifacts
9. Conclusion and Future Scope
10. References

1. Project Title :

**Visitor Identification and
Notification System**

2. Abstract :

In this paper we have proposed guest recognition system using face detection for security purpose. Implementation of the system is for monitoring whether any unknown person is entering in to the room of user. We are going to send notification containing name, date, photo on the user's phone.

For software coding tkinter, CV2, sklearn, numpy, pushbullet, face_recognition, pandas libraries are used. In order to get accurate and clear picture of an intruder we have used face_detection library for face detection. Assoon as the person enters near the door, pi camera captures the image and face detection process is done then if it matches with database images then a message with the picture of a person, name and date and time will be sent to theregistered mobile if person is not in data set then message with photo of unknown person will be sent.

In first methodology we used haar cascade classifier where wetake approx. 300 images of user using webcam and train the model and create.yml using LBPH classifier using opencv in python. Then whenever userclick on doorbell button it capture the user image and compare with each .yml that we have created if the guest image is matched then we can say that guest isknown otherwise guest is unknown.

In second methodology we used face_recognition library in which we take single clear image of guest and convert that image in ecoded list and store that list in program. Now whenever new user clicks on doorbell button it will capture the user image and convert into encoded list and compare with eachlist present in encodedKnownList if it match then send the notification containing name, date, time and image if not match the at position of name send“Unknown”.

Features Of Libraries :

1. Pandas :- Pandas is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library. Pandas is fast and it has high performance & productivity for users.

History: Pandas were initially developed by **Wes McKinney** in 2008 while he was working at AQR Capital Management. He convinced the AQR to allow him to open source the Pandas. Another AQR employee, **Chang She**, joined as the second major contributor to the library in 2012. Over time many versions of pandas have been released. The latest version of the pandas is 1.4.3.

Advantages of Pandas Library :

1. Data representation

- This helps to analyze and understand data better.
- Simpler data representation & better results for data science projects.

2. Less writing and more work done

- It is one of the best advantages of Pandas. What would have taken multiple lines in Python without any support libraries, can simply be achieved through 1-2 lines with the use of Pandas. Thus, using Pandas helps to shorten the procedure of handling data.

3. Efficiently handles large data

- Wes McKinney, the creator of Pandas, made the python library to mainly handle large datasets efficiently.
- Pandas help to save a lot of time by importing large amounts of data very fast.

Pandas package installation :

1) Open Installed anaconda prompt

2) Use the below command for package installation

```
pip install <packagename>
```

Ex: pip install pandas

3) Now, we can import the installed package into your program.

The major outcomes of the panda are:

- 1) Analysis of data
- 2) Preparation of data
- 3) Data manipulation
- 4) Data modeling
- 5) Data analysis

The major fields in which Python with Pandas is used are as below,

- 1) Finance
- 2) Economics
- 3) Analytics

2. 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.

Tkinter package installation

- 1) Open Installed anaconda prompt
- 2) Use the below command for package installation

```
pip install <packagename>
```

Ex: pip install tk

- 4) Now, we can import the installed package into your program.

- There are lost of alternative to Tkinter but among which Tkinter is most popular and most used librarie .

Ex : - Kivy, PySimpleGUI, Eel, DearPyGui

Advantages of using Tkinter

- - Tkinter is easy and fast to implement as compared to any other GUI toolkit.
- - Tkinter is more flexible and stable.
- - Tkinter is included in Python, so nothing extra need to download.
- - Tkinter provides a simple syntax.
- - Tkinter is really easy to understand and master.
- - Tkinter provides three geometry managers: place, pack, and grid. That is much more powerful and easy to use.

Disadvantages of using Tkinter

- - Tkinter does not include advanced widgets.
- - It doesn't have a reliable UI.
- - Sometime, it is hard to debug in Tkinter.
- - It is not purely Pythonic.

3. NumPy (Numerical Python) :

- 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.
- It is open-source software.

Why Use NumPy?

- 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.
- 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++.

NumPy package installation :

- 1) Open Installed anaconda prompt
- 2) Use the below command for package installation

`pip install <packagename>`

Ex: `pip install numpy`

- 3) Now, we can import the installed package into your program.

Advantages of NumPy:

- Numpy support some specific scientific functions such as linear algebra. They help us in solving linear equations.
- Numpy support vectorized operations, like elementwise addition and multiplication, computing Kronecker product, etc. Python lists fail to support these features.
- NumPy is very good for data analysis.
- It is a very good substitute for MATLAB, OCTAVE, etc as it provides similar functionalities and supports with faster development and less mental overhead(as python is easy to write and comprehend)

Why is NumPy Fast?

- Vectorization describes the absence of any explicit looping, indexing, etc., in the code - these things are taking place, of course, just “behind the scenes” in optimized, pre-compiled C code. Vectorized code has many advantages, among which are:
- vectorized code is more concise and easier to read
- fewer lines of code generally means fewer bugs

- the code more closely resembles standard mathematical notation (making it easier, typically, to correctly code mathematical constructs)
- vectorization results in more “Pythonic” code. Without vectorization, our code would be littered with inefficient and difficult to read for loops.

2.1 Tkinter messagebox :

- The messagebox module is used to display the message boxes in the python applications. There are the various functions which are used to display the relevant messages depending upon the application requirements.

- **Syntax:**

`messagebox.function_name(title, message [, options])`

Parameters :

- **function_name:** It represents an appropriate message box function.
- **title:** It is a string which is shown as a title of a message box.
- **message:** It is the string to be displayed as a message on the message box.
- **options:** There are various options which can be used to configure the message dialog box.
- **default:** This option is used to specify the default button like ABORT, RETRY, or IGNORE in the message box.

- **parent:** This option is used to specify the window on top of which the message box is to be displayed.

Function_Name:

- There are functions or methods available in the messagebox widget.
- `showinfo()`: Show some relevant information to the user.
- `showwarning()`: Display the warning to the user.
- `showerror()`: Display the error message to the user.
- `askquestion()`: Ask question and user has to answered in yes or no.
- `askokcancel()`: Confirm the user's action regarding some application activity.
- `askyesno()`: User can answer in yes or no for some action.
- `askretrycancel()`: Ask the user about doing a particular task again or not.

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.
- For working CSV files in python, there is an inbuilt module called csv.

CSV package installation

1) Open Installed anaconda prompt

2) Use the below command for package installation

```
pip install <packagename>
```

Ex: pip install python-csv

- Now, we can import the installed package into your program.

Advantages of CSV:

- CSV is human readable and easy to edit manually
- CSV is simple to implement and parse
- CSV is processed by almost all existing applications
- CSV provides a straightforward information schema
- CSV is faster to handle
- CSV is smaller in size
- CSV is considered to be standard format
- CSV is compact. For XML you start tag and end tag for each column in each row. In CSV you write the column headers only once.
- CSV is easy to generate

Disadvantages of CSV:

- CSV allows to move most basic data only. Complex configurations cannot be imported and exported this way.
- There is no distinction between text and numeric values.
- No standard way to represent binary data.
- Problems with importing CSV into SQL (no distinction between NULL and quotes).
- Poor support of special characters.
- No standard way to represent control characters.
- Lack of universal standard.

When Use CSV?

- When data has a strict tabular structure.
- To transfer large database between programs.
- To import and export data to office applications.
- To store, manage and modify shopping cart catalogue.

CSV File Characteristics :

- One line for each record
- Comma separated fields
- Space-characters adjacent to commas are ignored
- Fields with in-built commas are separated by double quote characters.
- Fields with double quote characters must be surrounded by double quotes. Each inbuilt double quote must be represented by a pair of consecutive quotes
- Fields that contain inbuilt line-breaks must be surrounded by double quotes

CV2 :

- Opencv is an open source library which is very useful for computer vision applications such as video analysis, CCTV footage analysis and image analysis. OpenCV is written by C++ and has more than 2,500 optimized algorithms.
- When we create applications for computer vision that we don't want to build from scratch we can use this library to start focusing on real world problems.
- There are many companies using this library today such as Google, Amazon, Microsoft and Toyota. Many researchers and developers contribute.
- We can easily install it in any OS like Windows, Ubuntu and MacOS.

CV2 package installation

- 1) Open Installed anaconda prompt
- 2) Use the below command for package installation

`pip install <packagename>`

Ex: `pip install opencv-python`

- Now, we can import the installed package into your program.

Key Features :

- face recognition
- Automated inspection and surveillance
- number of people – count (foot traffic in a mall, etc)
- Vehicle counting on highways along with their speeds
- Interactive art installations
- Anamoly (defect) detection in the manufacturing process (the odd defective products)
- Street view image stitching
- Video/image search and retrieval
- Robot and driver-less car navigation and control
- object recognition

- Medical image analysis
- Movies – 3D structure from motion
- TV Channels advertisement recognition

OpenCV Functionality :

- Image/video I/O, processing, display (core, imgproc, highgui)
- Object/feature detection (objdetect, features2d, nonfree)
- Geometry-based monocular or stereo computer vision (calib3d, stitching, videostab)
- Computational photography (photo, video, superres)
- Machine learning & clustering (ml, flann)
- CUDA acceleration (gpu)

3.1 Objective:

- Build smart home/office application for users
- To learn the skills of python and machine learning
- To provide a security option for user
- Get information about the guest/unknown person standing in front of camera.

3.2 Introduction :

Nowadays, as the technology is increasing, facilities for human beings are increasing. In day to day activities, life of people has become very easier with the incorporation of many technologies. On the other hand, it also creates security issues. The traditional door locks have a problem that almost anyone can break and enter into your house. Hence it is a great challenge to overcome these problems. In general, in order to secure home, people make use of CCTV. Images will store in the database, so that the action can be taken when any suspicious incident happens. This type of approach is a passive. But there is a need for an active approach. This type of approach is nothing but where actions can be taken immediately as soon as a security threat occurs.

Hence, Face is the representation of one's unique identity. So here proposing an application using face recognition for purpose of identifying the guest and sending notification on user's mobile phone.

Face recognition is done in 2 steps:

1. Feature extraction
2. Face matching

1.FEATURE
EXTRACTION:

After the previous two steps, feature extraction is performed resulting in effective information that is useful for distinguishing between faces of different persons and stable with respect to the geometrical and photometrical variations.

2. FACE MATCHING:

The extracted features are compared to those stored in the database, and decisions are made according to the sufficient confidence in the match score.

At the end of project we will be able to perform following tasks:

1. Recognition of guest/unknown person
2. Send notification to user

3.3 Literature survey :

Mayur Surve et al. in 2020, have developed a model which catches the live images from Camera. Then it applies different algorithms for face detection and [face recognition](#). They also created the GUI on single click which catches the images, forms the dataset and inculcates the dataset. They used the Haar cascade algorithm to recognize the face in the image.

Palanivel Net al. in 2019, has approached a module that signs the existence of people by detecting their facial nature and creating the attendance data itself.

Face Recognition's reliability change is bestrewed with the aspects similar to changing the glowing, posture changes, expression changes, and occlusion. They used K-means clustering algorithmic rule for analyzing the face characteristics. Biometric nature of the face features has been withdrawn. The K-mean clustering approach is used in gathering the face characteristics. Then, SVM method is used in detection of the photos characteristics. It may fulfill high identification showing with lesser characteristics number.

Jenif D Souza W Set al. in 2019, has proposed a system where image processing techniques is used for facial recognition. The refined photo is utilized to compare with the stored catalogue. The commenced procedure was being effectuated with 4 modules such as Image Capturing, Cleavage of group photo and Face Detection, Face comparison and Recognition.

AZM Ehtesham Chowdhury et al. in 2019, have proposed a rare camera prototype to analyze the attendance much effectively. For developing a model with additional vigorous and steady. A different algorithm was also put forwarded to exercise the technique perfectly. This technique will make use of a modus to analyze student's attendance. This is based on face recognition and detection. Precision in average was the basic interest for selecting the most valid modus.

Nandhini R, Duraimurugan Net al. in 2019, has built a technique which is capable to detect and acknowledge student faces quickly and accurately in photos or videos which is then detained via a camera. Many algorithms and technologies have been implemented for tweaking the execution of recognizing the face using Deep Learning.

M. Kasiselvanathan et al. in 2018, has built a technique named automatic attendance management system by face detection technique. The system used to recognize the facial dimensions in order to detect the face. An effective face recognition system has been implemented by upgrading the quality of the

system. Eigen Faces algorithm has been used in this system. The technique is not only recognizing the faces, also the space of the facial nature based on changing rules.

Omar Abdul Rhman Salim et al. in 2018, had proposed a technique of implementing a fully implanted student attendance process by face detection. The technique is depending on Raspberry Pi which runs Raspbian Operating System. The Camera and a 5-inch screen is connected to the Raspberry Pi. The image captured from the camera will be transferred to the Raspberry Pi. Which is internally programmed to handle face recognition by developing the LBPs. If the face in the input image i.e., image taken matches with the trained dataset image, the door will be opened and the attendance will be taken positively and it will be stored

4. Problem statement :

To design guest recognition system using face detection and send the notification to the user of system containing guest image, data and name of guest if present in data-set, if guest data is not present in data-set then at position of name send 'Unknown' and send image capture in image video, send time and date when image is get captured.

5. Methodology:

Approach 1. Using Haar Cascade Algorithm:

5.1.1 Overview:

Haar cascades is a machine learning based approach where a cascade function is trained with a set of input data. Haar Cascade is a machine learning object detection algorithm used to identify objects in an image or video. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

5.1.2 Face detection:

In the field of technology Face detection is treated as the demanding and practically applied approach. The identification of each face present in an image is the major task of the face detection, Here the implementation is done using OpenCV.

- i. Loading the input images.
- ii. Converting the input images into [gray scale](#) images.
- iii. Applying the Haar cascade and [LBP](#) classifier.
- iv. Comparing both classifier based on the accuracy and time.

- A. Importing the required libraries
- B. Taking the images which are captured by the camera.
- C. To process the image through the classifiers it is converted into grayscale image.
- D. Image will be loaded using OpenCV
- E. By default, image will be loaded into BGR color space

5.1.3 Haar cascade classifier

- i. Loading the input image using built in function `cv2.imread(img_path)`, here the passing the image path as an input parameter
- ii. Converting it to gray scale mode and then displaying it

iii. Loading the haar cascade classifier

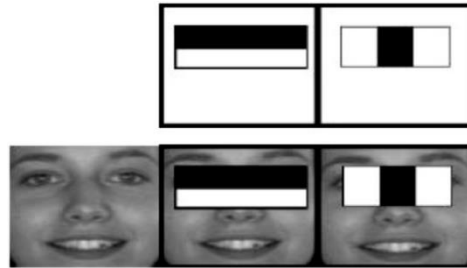
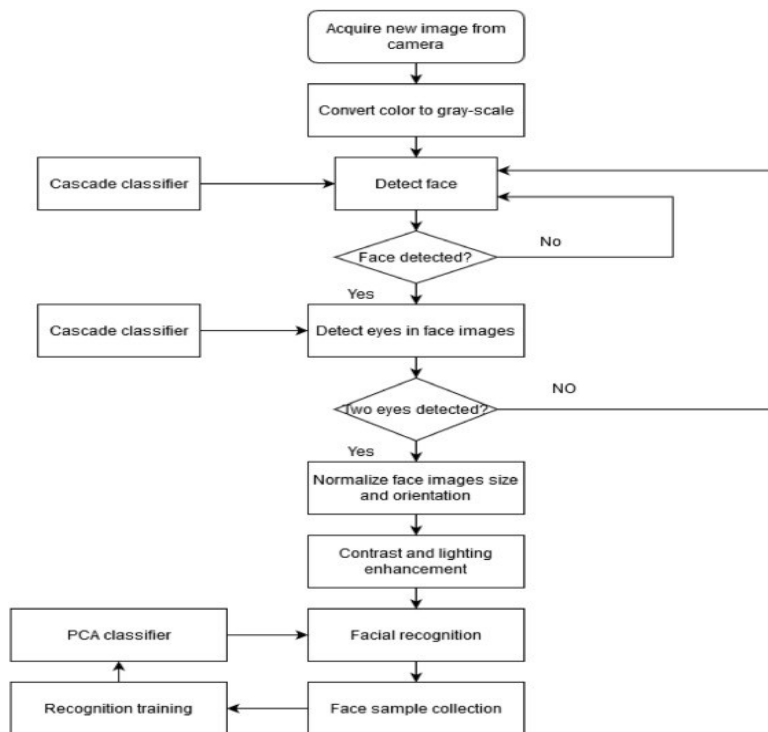


Fig.1 represents the Haar like feature. It consists of edge feature and linefeature. In the gray-scale image the white bar represents the pixels that are closer to the light source

Flow Chart:



Approach 2. Using face recognition library:

5.2.1 Overview:

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

a. Find all faces in pictures



Input



Output

import face_recognition

```
image=face_recognition.load_image_file("your_file.jpg")
```

```
face_locations = face_recognition.face_locations(image)
```

Find and manipulate facial features in pictures. Get the locations and outlines of each person's eyes, nose, mouth and chin. Finding facial features is super useful for lots of important stuff



Input



Output

import face_recognition

```
image = face_recognition.load_image_file("your_file.jpg")
```

```
face_landmarks_list = face_recognition.face_landmarks(image)
```

a. Identify faces in pictures. Recognize who appears in each photo.



Input



Picture contains
"Joe Biden"

Output

import face_recognition

```
known_image = face_recognition.load_image_file("biden.jpg")
```

```
unknown_image = face_recognition.load_image_file("unknown.jpg")
```

```
biden_encoding = face_recognition.face_encodings(known_image)[0]
```

```
unknown_encoding = face_recognition.face_encodings(unknown_image)
```

```
results = face_recognition.compare_faces([biden_encoding], unknown_encoding)
```

Then use Pushbullet (Pushbullet, a prominent Python package, which connects multiple devices using python code) for sending notification to the user of system containing image of guest, date and name of guest.

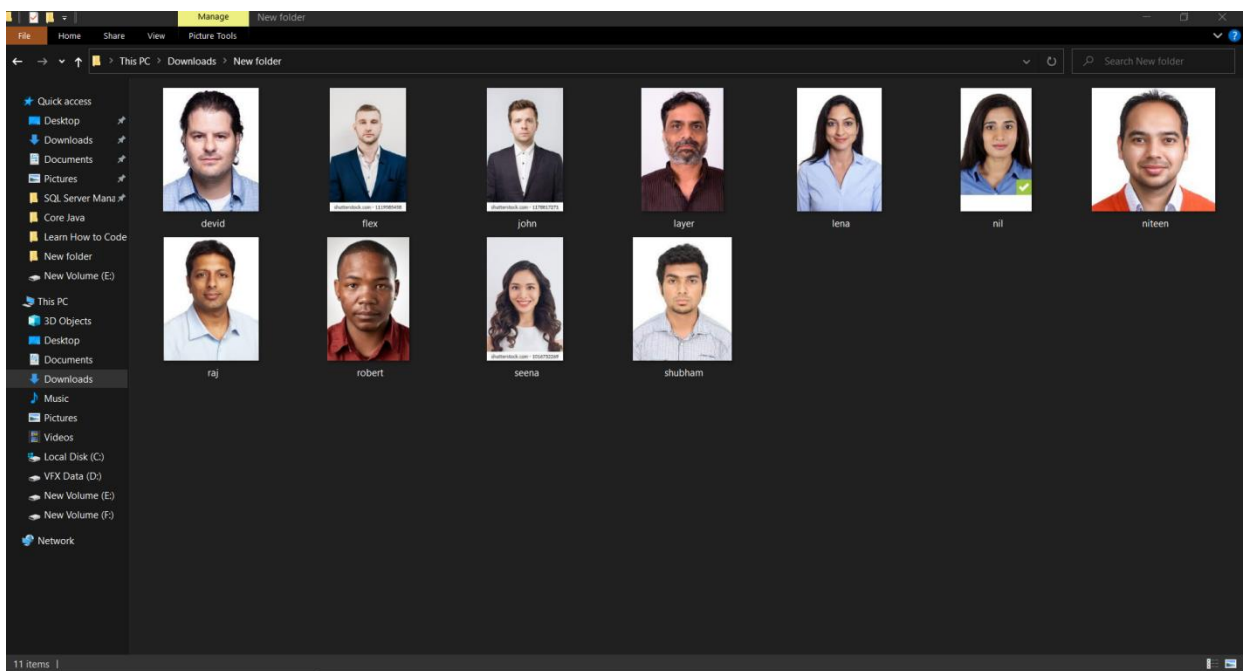
Using our computer and python code, we will send some messages to the pushbullet app installed on a mobile device, which has the same login id, as the one on the computer. For this first, we'll need to create an account on *Pushbullet.com* and sign in to it, from computer and phone both.

```
access_token="o.ujPzax3LopNgYH91xWIwVMRpzTk6t6e3"
def sendNotification(personName, personImg):
    # Get the instance using access token
    pb = PushBullet(access_token)
    # Send the data by passing the main title and text to be sent
    pb.push_note("Hello, how are you")
    with open(personImg, "rb") as pic:
        file_data = pb.upload_file(pic, personImg)
    push = pb.push_file(**file_data)
    # push2 = pb.push_file("Person", personImg, file_type="image/jpeg")
    print("Message sent successfully...")
```


6. Testing:



In testing perform operations like deletion of record, insertion of new images and sending notification for correct guest. We take 20 guest images and store that images in folder and check working of notification functionality.



We store high resolution images in Images_Attendance folder with name of that person.

Format of Encoded Image:

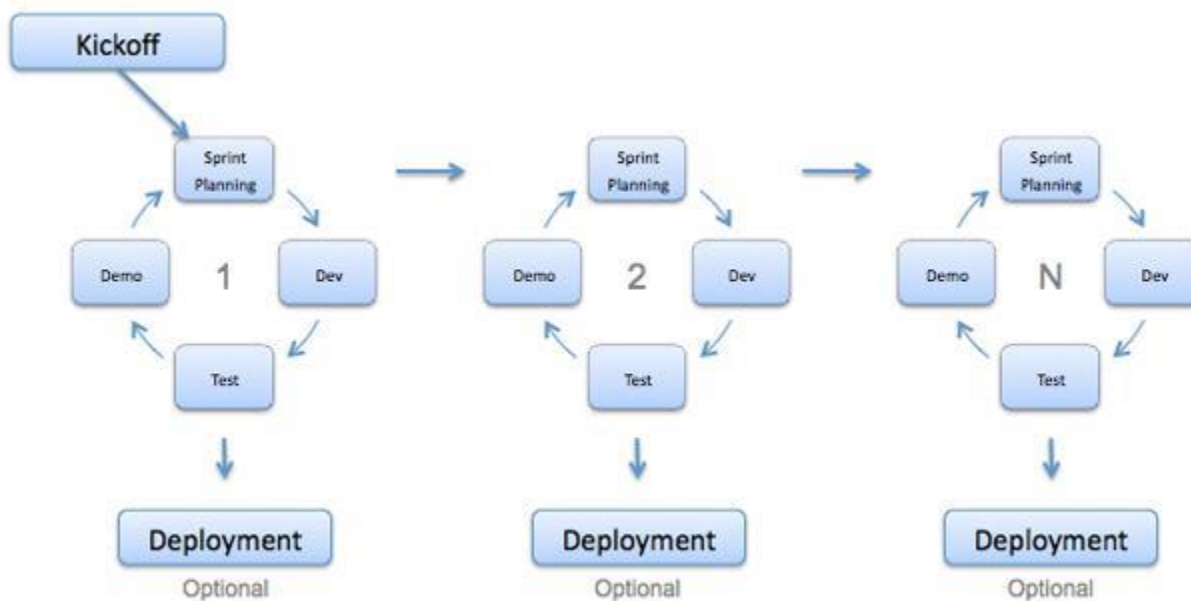
FACE ID: [0.74648741 0.7896611 0.58822812 0.8228576 0.67038206
0.7097386 0.43233329 0.60983211 0.86816181 0.63893744 0.67676455
0.672946 0.78353032 0.76080103]

7. SDLC Model Used for the Project :

What is Agile model – advantages, disadvantages and when to use it?

Agile development model is also a type of Incremental model. Software is developed in incremental, rapid cycles. This results in small incremental releases with each release building on previous functionality. Each release is thoroughly tested to ensure software quality is maintained. It is used for time critical applications. Extreme Programming (XP) is currently one of the most well known agile development life cycle model. Agile methodology is explained in more detail under the Agile Tester topics.

Diagram of Agile model:



Advantages of Agile model:

- Customer satisfaction by rapid, continuous delivery of useful software.
- People and interactions are emphasized rather than process and tools. Customers, developers and testers constantly interact with each other.
- Working software is delivered frequently (weeks rather than months).
- Face-to-face conversation is the best form of communication.
- Close, daily cooperation between business people and developers.
- Continuous attention to technical excellence and good design.
- Regular adaptation to changing circumstances.
- Even late changes in requirements are welcomed

Disadvantages of Agile model:

- In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.
- There is lack of emphasis on necessary designing and documentation.

- The project can easily get taken off track if the customer representative is not clear what final outcome that they want.
- Only senior programmers are capable of taking the kind of decisions required during the development process. Hence it has no place for newbie programmers, unless combined with experienced resources.

When to use Agile model:

- When new changes are needed to be implemented. The freedom agile gives to change is very important. New changes can be implemented at very little cost because of the frequency of new increments that are produced.
- To implement a new feature the developers need to lose only the work of a few days, or even only hours, to roll back and implement it.
- Unlike the waterfall model in agile model very limited **planning** is required to get started with the project. Agile assumes that the end users' needs are ever changing in a dynamic business and IT world. Changes can be discussed and features can be newly effected or removed based on feedback. This effectively gives the customer the finished system they want or need.
- Both system developers and stakeholders alike, find they also get more freedom of time and options than if the software was developed in a more rigid sequential way. Having options gives them the ability to leave important decisions until more or better data or even entire hosting programs are available; meaning the project can continue to move forward without fear of reaching a sudden standstill.

Waterfall model:

It is Old SDLC Model. The Waterfall Model was the first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

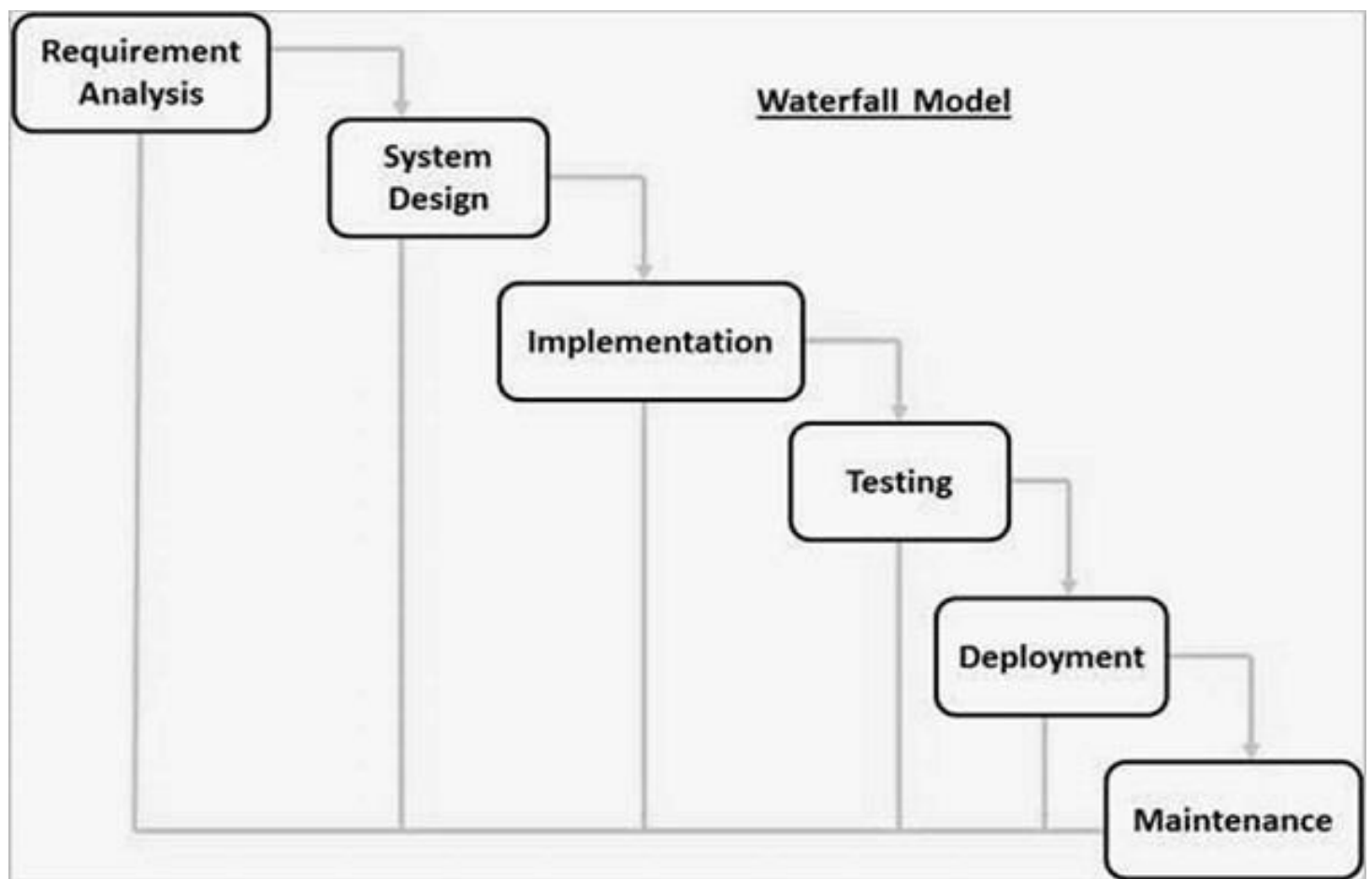
The Waterfall model is the earliest SDLC approach that was used for software development.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

Waterfall Model - Design

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

The following illustration is a representation of the different phases of the Waterfall Model.



Waterfall Model - Design

The sequential phases in Waterfall model are –

- **Requirement Gathering and analysis** – All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** – The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation** – With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing** – All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** – Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** – There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model, phases do not overlap.

Waterfall Model - Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are –

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

Waterfall Model - Advantages

The advantages of waterfall development are that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.

Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

Some of the major advantages of the Waterfall Model are as follows –

- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.

- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

Waterfall Model - Disadvantages

The disadvantage of waterfall development is that it does not allow much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.

The major disadvantages of the Waterfall Model are as follows –

- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty is high with this process model.
- It is difficult to measure progress within stages.
- Cannot accommodate changing requirements.
- Adjusting scope during the life cycle can end a project.
- Integration is done as a "big-bang" at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early.

8. Result and Analysis:

Home page of application :

We are able to capture guest image and identify whether the guest is known or unknown and after that send the notification to the user of system.

Below are screenshot of our system:

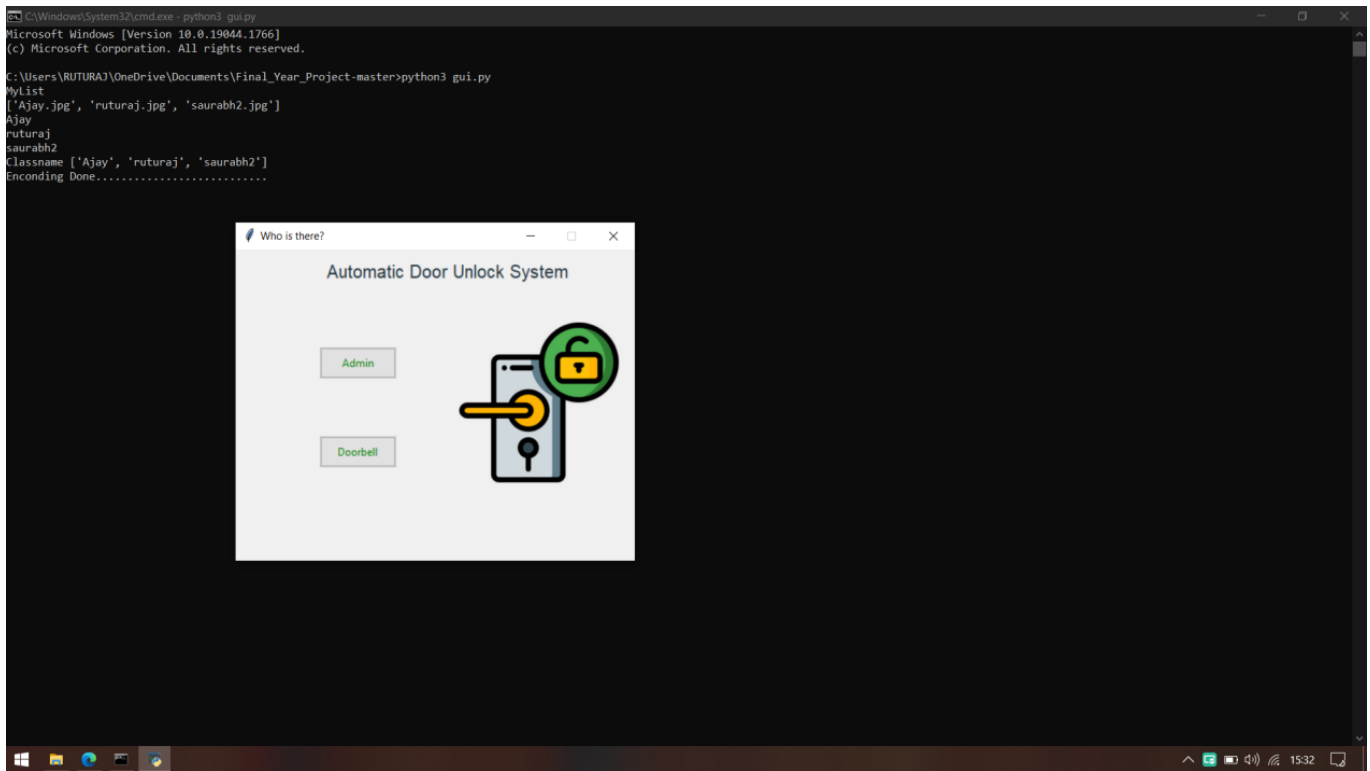


Fig.1.1 Home page of application

Admin Portal :

At starting of GUI we have 2 buttons one is for Admin and another is for guest. In Admin section we can add new guest record by uploading image or capturing image, we can also delete record and display list of guest.

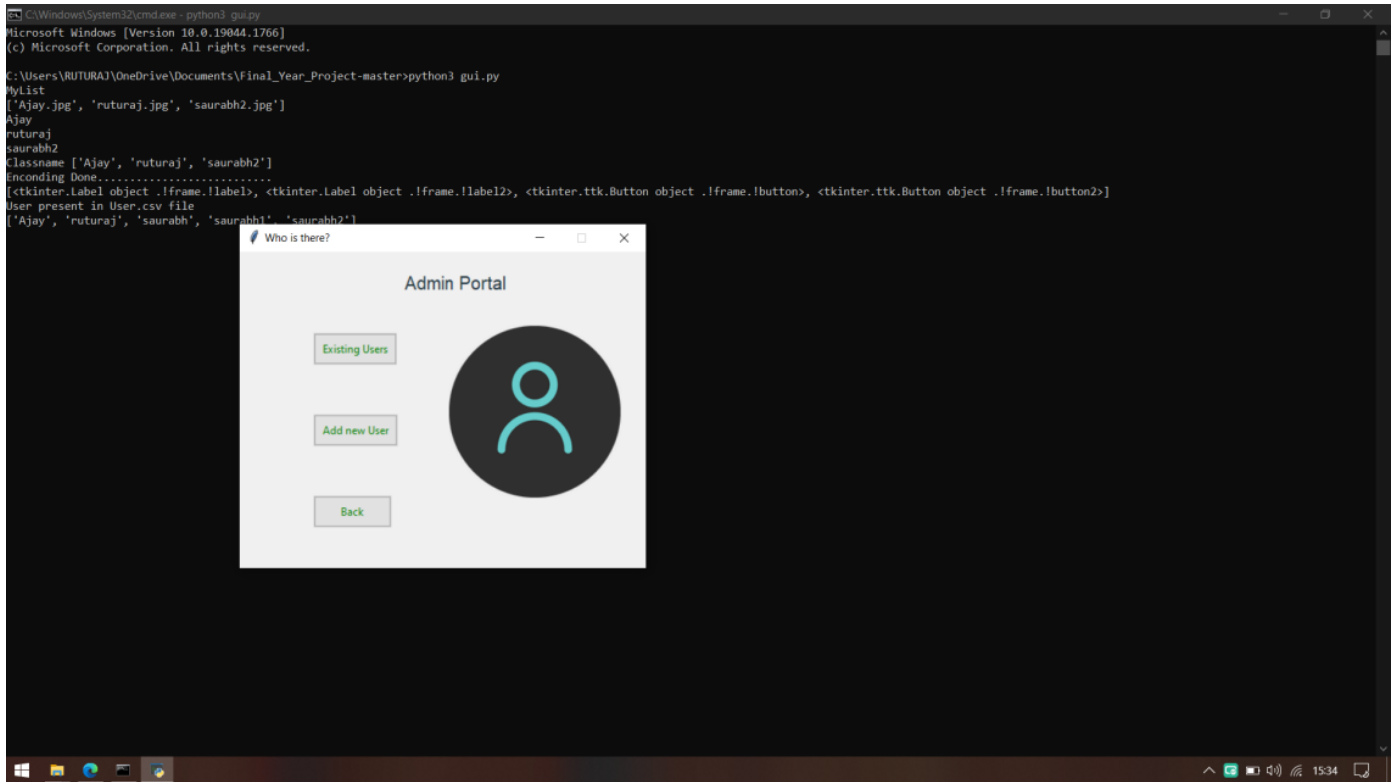


Fig.1.2. Admin Portal

In admin section we have three buttons one for checking list of available users, second is for addition of new user and third is for going back to the main frame.

List of guest images present :

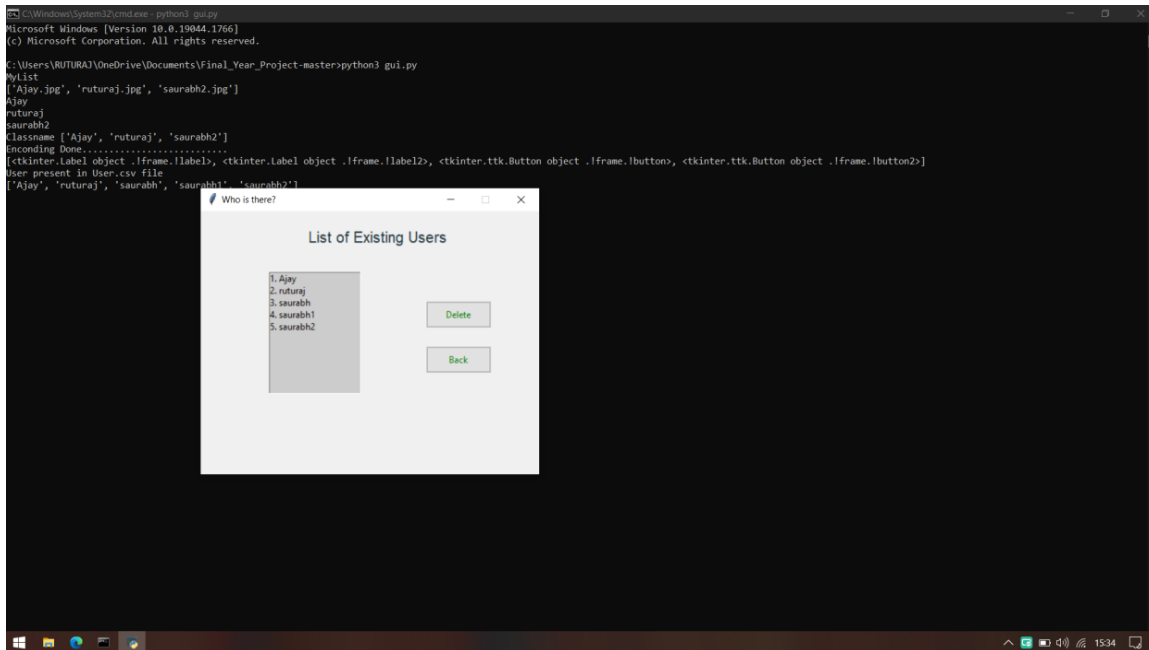


Fig.1.3. Show list of guest images present

After clicking on exiting users button it shows the list of all available guest name,we can also delete guest record by clicking on its name then click on deletebutton.

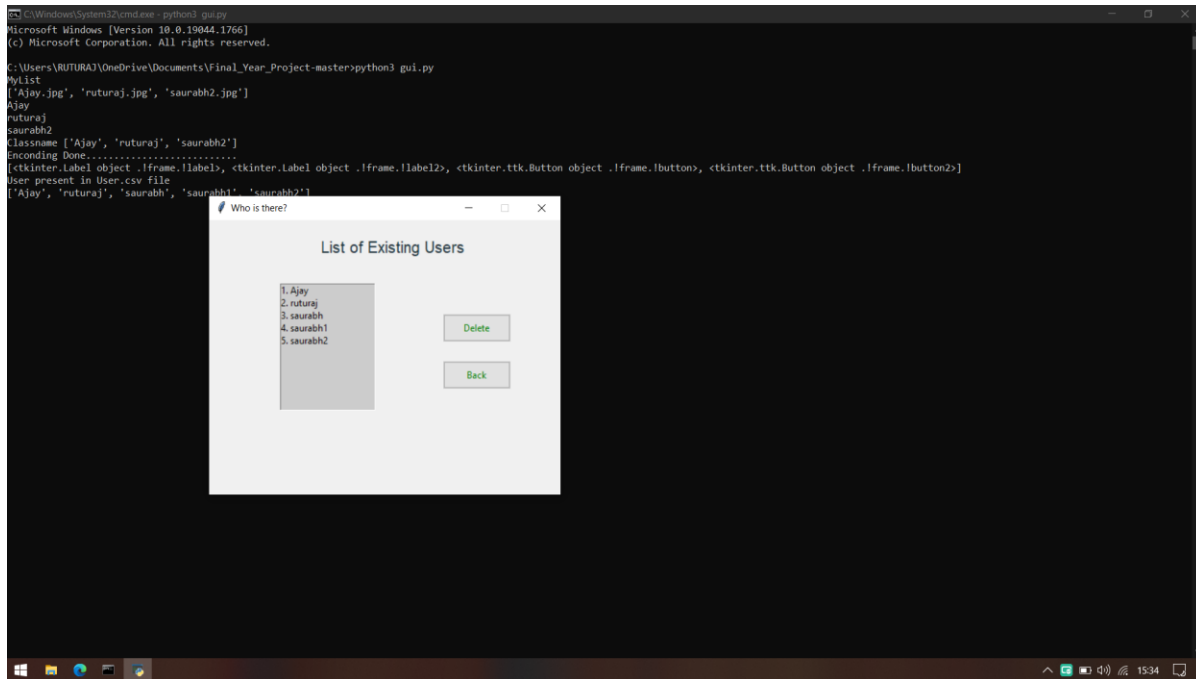


Fig.1.4. Get user user name and image

In order to add new guest you have enter unique guest name(i.e. name not already exists) then after we can capture or upload the image of guest. After that is shows the pop-up messages 'Imaged Saved' shown in below image.

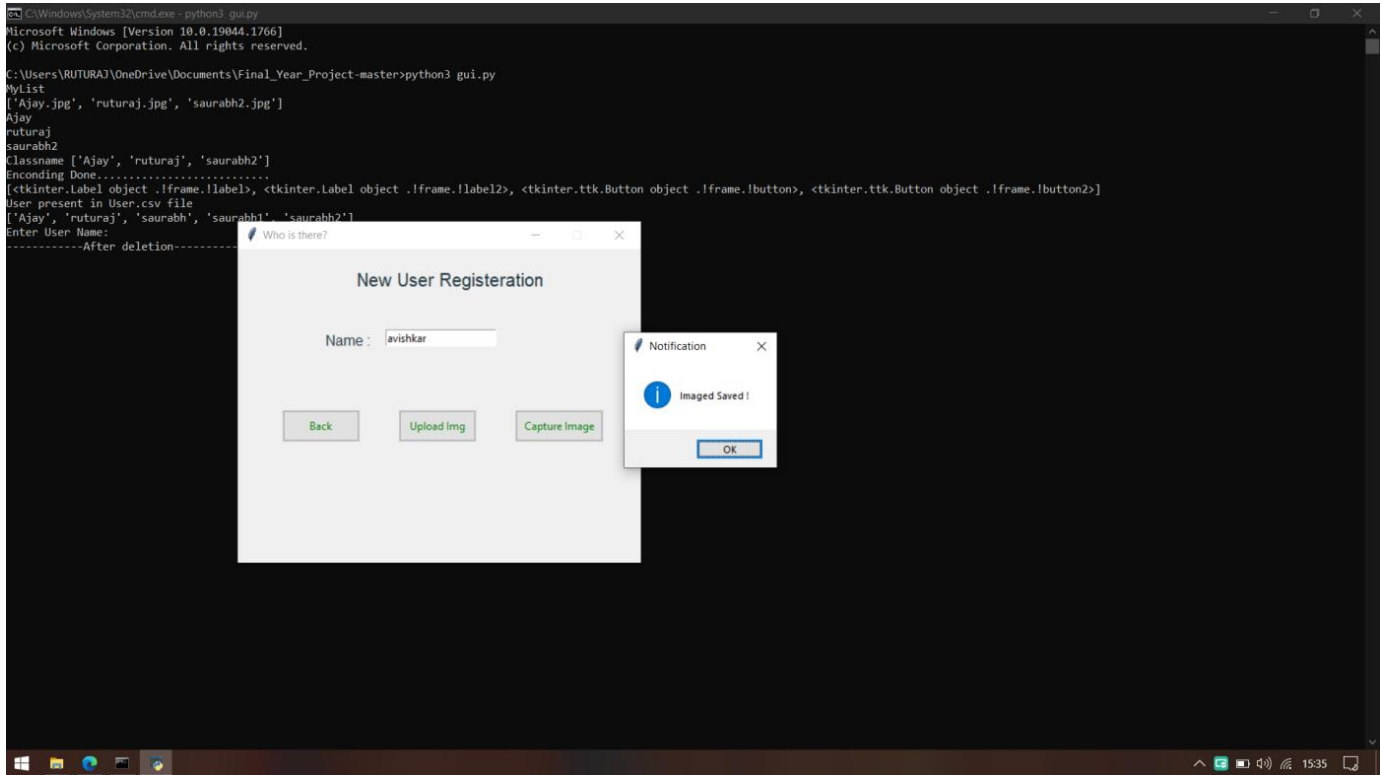


Fig.1.5. Show pop-up message after image saved in folder

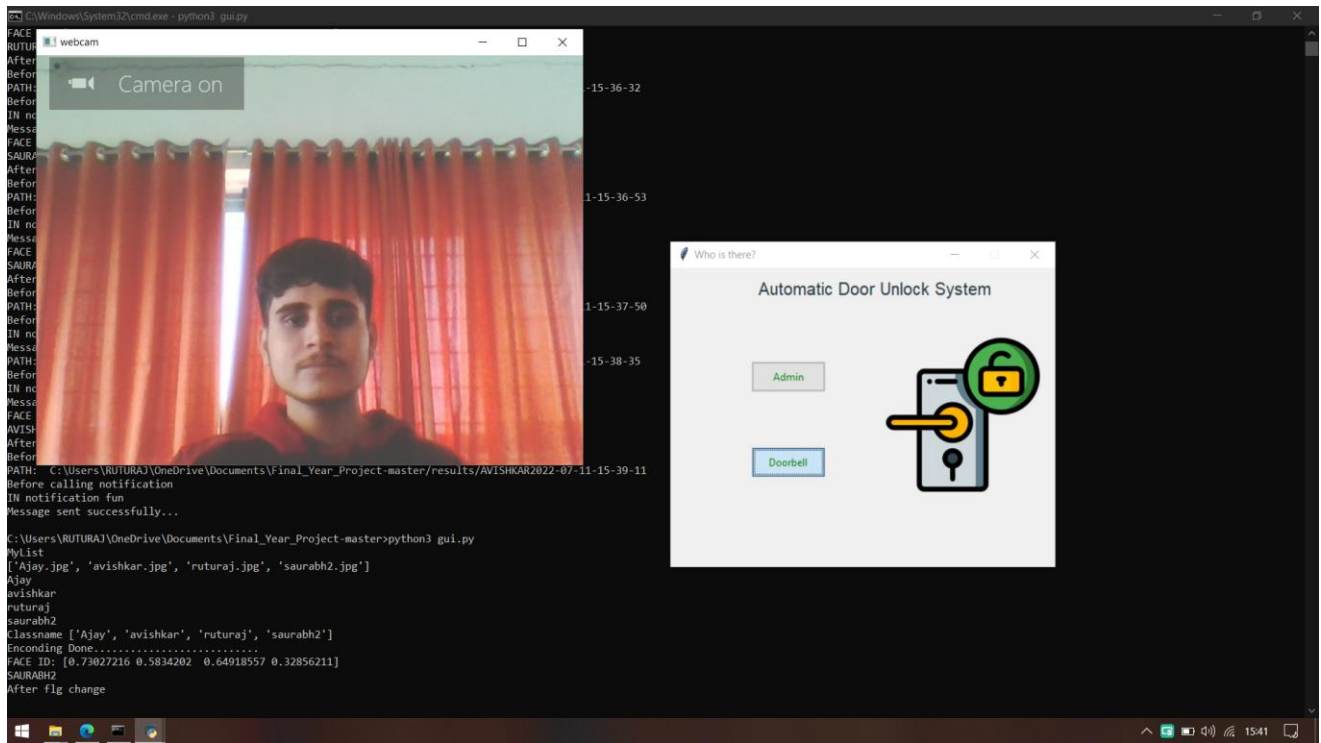


Fig.1.6.Click on doorbell button in order to check person known or unknown

In main frame on clicking on doorbell button web-cam gets open and check for the guest is known or unknown and send the notification on push-bullet application. Below is an example of how messages will appear in application.

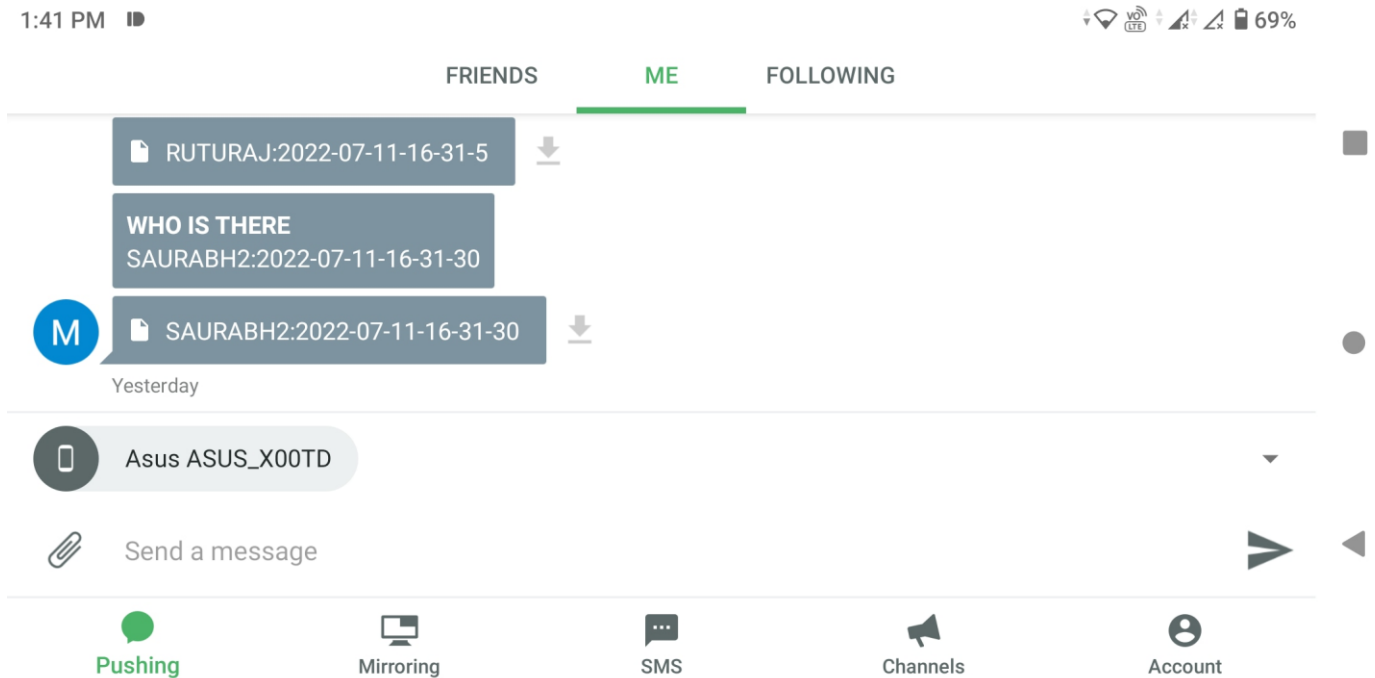


Fig.1.7. Send notification on system owner mobile

9. Performance Analysis:

- While testing the application with authorized person's face it detected and recognized user with rate of 100%.
- In good lighting conditions (Daylight) it gave recognition with accuracy of 92.77 %.
- While in low light conditions it gave recognition with accuracy of 77%.
- Although the profile is just for frontal faces it can detect accurately for.
- slight head tilts (about 15 -20 degrees) and at 200 cms distance from webcam.
- The accuracy varies between lighting conditions and the quality of the webcam used.
- For recognition with even greater confidence, the equipment or environment (web camera and the room light/natural light) should be higher quality.

10. Project Artifact:

Program Flow chart:

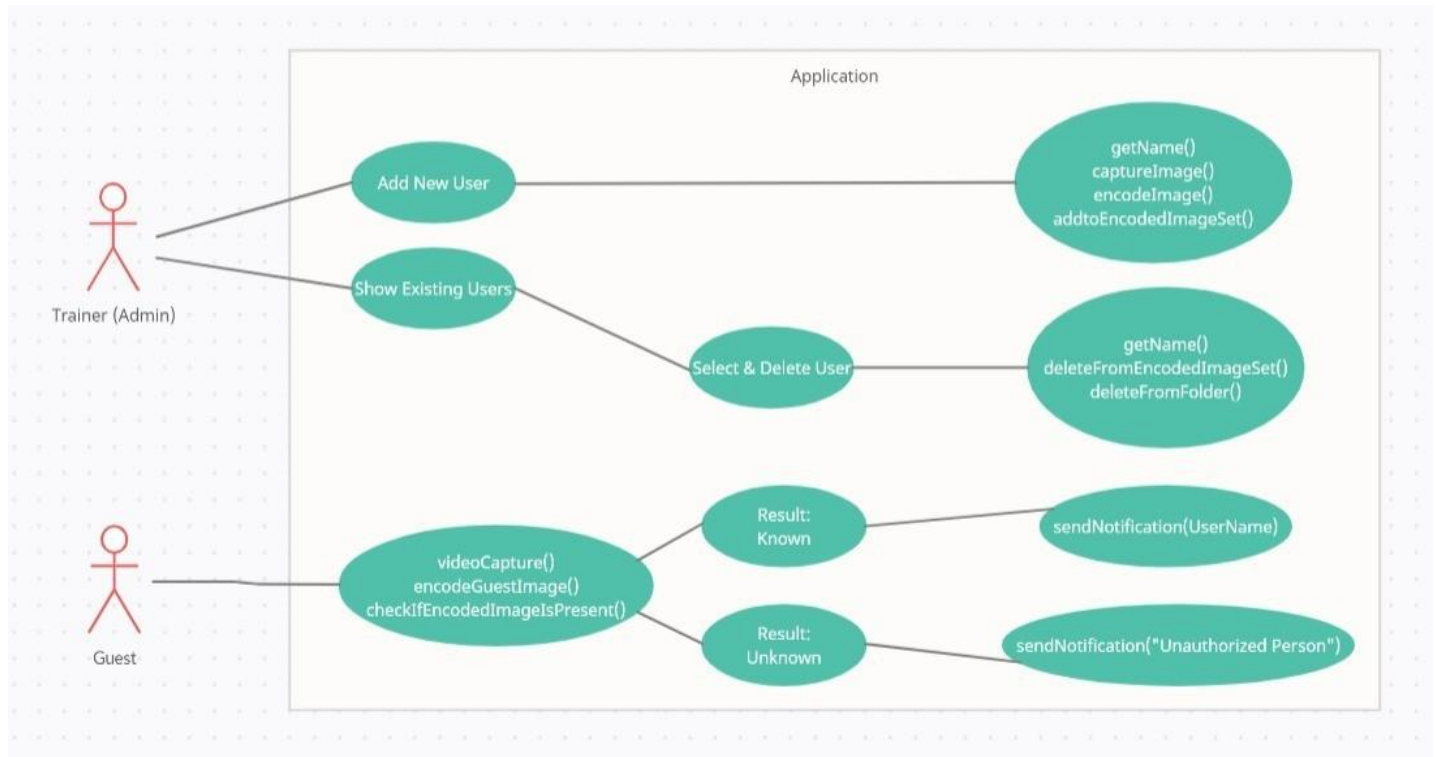


Fig.2.1 Flow of system

11.Gantt Chart:



WHOE IS THERE

GANTT CHART



Project code:

<https://github.com/ruturaj3123/visitor-identification-and-notification/tree/main/Project>

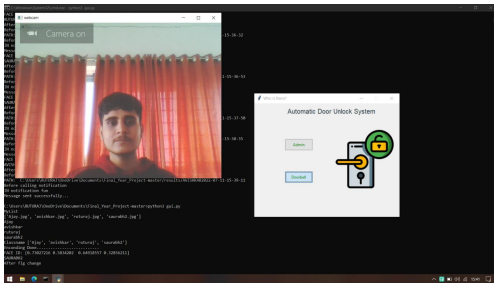
12. Test Cases:

Take 10 images of different people and saved images in data-set and checked for person is known or unknown after testing get all person are known who's image store in data-set and send the successfully notification to the owner of system with that person name, time, date and image.

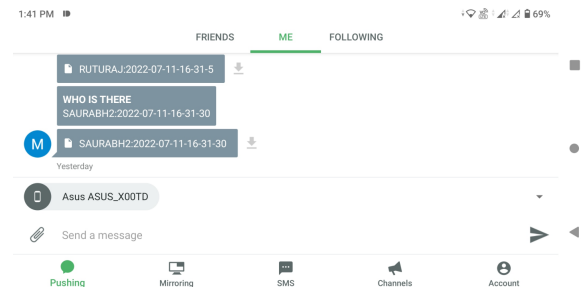
Secondly we checked for 10 unknown person out 10 person 9 persons get detected as unknown and 1 get detected as known with other's name,
Sending notification is successfully done in both cases.

Test Case1:

For Known Person: Image of person present in folder so send notification with name of that person.



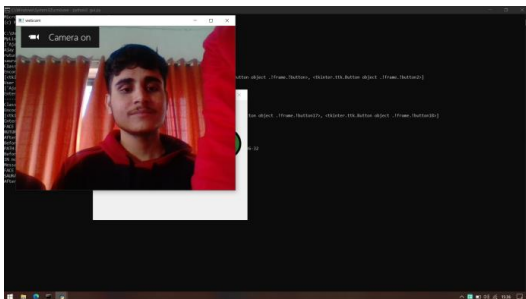
Capturing image



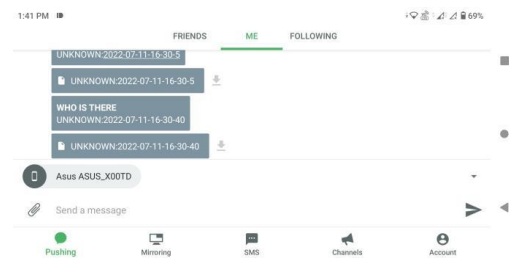
end result(known)

Test Case 2:

For Unknown Person: Image of person is not present folder so send notification with name as "Unknown".



Capturing image



end result(unknown)

13.Conclusion and future scope :

Conclusion :

- The Accuracy of for the algorithm is very high, which shows that theLBPH classifier is an efficient and accurate face recognizer.
- The algorithm is able to provide access control during different lighting conditions (during day and night) because Haar Cascades frontal profile.
- Although the profile is just for frontal faces it can detect accurately forslight head tilts (about 15 - 20 degrees) and at 200 cms distance from webcam.
- The accuracy between lighting conditions and the quality of the webcam used.
- For recognition with even greater confidence, the equipment or environment (web camera and the room light/natural light) should behigher quality.
- face_recognition library used in second approach built using dlib's state-of-the-art face recognition built with deep learning. The model has anaccuracy of 99.38% on the [Labeled Faces in the Wild](#) benchmark.

14.Future Scope :

- This project can be integrated with raspberry pi for actual implementation on the door.
- This project can be used in drone cams for criminal detection in crowds.
- This project can be implemented over large sectors.
- Can be used for attendance purpose of students.
- Can be used for employee recognition in corporate sectors.
- Can be used to detect security flaws and trespassers.

Code :

```
from fileinput import filename

from itertools import count

from sre_constants import SUCCESS

import tkinter as tk

from tkinter import messagebox

from tkinter import ttk

from PIL import ImageTk, Image

from cv2 import imshow

from sklearn.neighbors import NearestCentroid

from getkey import getkey, key

import pytesseract

import cv2

import numpy as np

import face_recognition

import pandas as pd

import os

import csv

import time

import datetime

import sys

import shutil

from pushbullet import PushBullet

from pywebio.input import *

from pywebio.output import *

from pywebio.session import *

from tkinter import filedialog

from tkinter.filedialog import askopenfile

access_token="o.fVGtMeajD1mIFzDei8cH8UPYWsYp0omo"
```

```
path = 'Images_Attendance'
```

```

images = []

classNames = []

deleted =[]

myList = os.listdir(path)

print("MyList")

print(myList)

for cl in myList:

    curImg = cv2.imread(f'{path}/{cl}')

    images.append(curImg)

    print(os.path.splitext(cl)[0])

    classNames.append(os.path.splitext(cl)[0])

print("Classname",classNames)

```

```

def findEncodings(images):

    encodeList =[]

    for img in images:

        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

        encode = face_recognition.face_encodings(img)[0]

        encodeList.append(encode)

    print("Encoding Done.....")

    return encodeList

```

```

encodeListKnown=findEncodings(images)

```

```

def startpage(container):

    label = tk.Label(container, text ="Automatic Door Unlock System", font = "Helvetica", foreground="#263942")    #Add
    label

    label.config(font=("Helvetica", 15))    #set label size and font

    label.place(x = 100,y = 10)

    def admin_clear_frame(frame):

        print(frame.winfo_children())    #get list of all children widgets

        for widget in frame.winfo_children():

            widget.destroy()    #removing widget

```

```
admin(frame)    #calling admin function with empty frame as argument
```

```
# opens the image

img = Image.open('static/door.png')

img = img.resize((180, 180), Image.Resampling.LANCZOS)

# PhotoImage class is used to add image to widgets, icons etc

img = ImageTk.PhotoImage(img)

# create a label

panel = tk.Label(container, image = img)

# set the image as img

panel.image = img

panel.place(x = 250 , y = 80)    #place the door image

ttk.Style().configure("TButton", padding=6, relief="flat",
                        background="#ccc",foreground='green')
```

```
button1 = ttk.Button(container, text ="Admin",command = lambda : admin_clear_frame(container))    #call
admin_clear_frame function on click

button1.place(x = 95, y = 110)

button2 = ttk.Button(container, text ="Doorbell",command = lambda : doorbell(button1,button2))    #call doorbell function
to check authorised user or not

button2.place(x = 95,y = 210)
```

```
def admin(container):

    label = tk.Label(container, text ="Admin Portal", font = "Helvetica", foreground="#263942")

    label.config(font=("Helvetica", 15))

    label.place(x = 180,y = 20)

    img = Image.open('static/login.png')

    img = img.resize((190, 190), Image.Resampling.LANCZOS)

    img = ImageTk.PhotoImage(img)
```



```
panel = tk.Label(container, image = img)

panel.image = img

panel.place(x = 230 , y = 80)

ttk.Style().configure("TButton", padding=6, relief="flat",
                      background="#ccc",foreground='green')
```

```
def user_list_clear_frame(frame):

    for widget in frame.winfo_children():    #clearing frame

        widget.destroy()

    user_list(frame)    #call user_list with empty frame
```

```
def new_user_clear_frame(frame):

    for widget in frame.winfo_children():    #clearing frame

        widget.destroy()

    new_user(frame) #call new_user with empty frame
```

```
def back_menu(frame):

    for widget in frame.winfo_children():    #clearing frame

        widget.destroy()
```

```
startpage(frame)    #calling startpage
```

```
button1 = ttk.Button(container, text ="Existing Users",command = lambda : user_list_clear_frame(container)) #call
user_list_clear_frame on click

button1.place(x = 82, y = 90)

button2 = ttk.Button(container, text ="Add new User",command = lambda : new_user_clear_frame(container)) #call
new_user_clear_frame on click

button2.place(x = 82,y = 180)
```

```
button3 = ttk.Button(container, text = "Back", command = lambda : back_menu(container)) #call back_menu on click

button3.place(x = 82, y = 270)
```

```
def new_user(container):

    new_user = tk.StringVar()

    flag = tk.IntVar()

    flag.set(0)

    num_images = tk.IntVar()
```

```
label = tk.Label(container, text = "New User Registration", font = "Helvetica", foreground = "#263942")

label.config(font = ("Helvetica", 15))

label.place(x = 130, y = 20)

name_label = tk.Label(container, text = "Name :", font = "Helvetica", foreground = "#263942")

name_label.config(font = ("Helvetica", 12))

name_label.place(x = 95, y = 90)
```

```
def clear(frame):

    for widget in frame.winfo_children():

        widget.destroy()

    admin(frame)
```

```
def check(container, name):

    if(name == ""):

        return

    data = pd.read_csv('User.csv')

    if(name in list(data.Name)):

        messagebox.showerror("Error", "User Name already Exists")

    return
```

```
if(name.upper() in deleted):

    deleted.remove(name.upper())
```

```
print("-----After deletion-----",deleted)

data = pd.read_csv('User.csv')

data.loc[len(data.Name)] = [name]

data.set_index('Name',inplace=True)

data.to_csv('User.csv')
```

```
path2=os.getcwd() + f"/Images_Attendance/"

path2=path2+str(name)

vid = cv2.VideoCapture(2)

while(True):

    # Capture the video frame

    # by frame

    ret, frame = vid.read()

    # Display the resulting frame

    cv2.imshow('Frame', frame)

    # the 'q' button is set as the

    # quitting button you may use any

    # desired button of your choice

    if cv2.waitKey(1) & 0xFF == ord('q'):

        cv2.imwrite(path2+".jpg",frame)

        break

# After the loop release the cap object

vid.release()

# Destroy all the windows

cv2.destroyAllWindows()

# encodeListKnown=findEncodings()

messagebox.showinfo("Notification","Imaged Saved !")
```

```
clear(container)

curImg = cv2.imread(path2+".jpg")

images.append(curImg)

classNames.append(name)

print("Classname",classNames)
```

```
curImg = cv2.cvtColor(curImg, cv2.COLOR_BGR2RGB)

encode = face_recognition.face_encodings(curImg)[0]

encodeListKnown.append(encode)

print("Encodeing done after image upload")
```

```
def upload_img(container,name):

    if(name==""):

        print("Empty user name")

        return

    data = pd.read_csv('User.csv')

    if(name in list(data.Name)):

        messagebox.showerror("Error","User Name already Exists")

        return
```

```
if(name.upper() in deleted):

    deleted.remove(name.upper())

print("-----After deletion-----",deleted)

data = pd.read_csv('User.csv')

data.loc[len(data.Name)] = [name]

data.set_index('Name',inplace=True)

data.to_csv('User.csv')
```

```
f_type=[('Jpg Files','*.jpg')]

file_name=filedialog.askopenfilename(filetypes=f_type)
```

```
img=cv2.imread(file_name)

path2=os.getcwd() + f"/Images_Attendance/"

path2=path2+str(name)

cv2.imwrite(path2+".jpg",img)

cv2.destroyAllWindows()
```

```
messagebox.showinfo("Notification","Imaged Saved !")

clear(container)
```

```
images.append(img)

classNames.append(name)

print("Classname",classNames)
```

```
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

encode = face_recognition.face_encodings(img)[0]

encodeListKnown.append(encode)

print("Encodeing done after image upload")
```

```
entry_name = tk.Entry(container,textvariable = new_user)    #take user name and save to new_user

print("Enter User Name:", new_user.get())

entry_name.place(x = 165, y = 90)

ttk.Style().configure("TButton", padding=6, relief="flat",background="#ccc",foreground='green')


button3 = ttk.Button(container, text ="Back",command = lambda : clear(container),state = tk.NORMAL) #adfter click go
back to admin frame
```

```
button2 = ttk.Button(container, text ="Upload Img",command = lambda : upload_img(container,new_user.get()))

#Button to create dataset

button1 = ttk.Button(container, text ="Capture Image",command = lambda : check(container,new_user.get()))
```

```
button1.place(x = 310, y = 180)

button2.place(x = 180,y = 180)

button3.place(x = 50,y = 180)
```

```
def delete_selected(frame,Lb1): #delete selected user releted data

    a = Lb1.get(Lb1.curselection()).split(' ')

    print(a)    #['1.', 'Ajay']
```

```
name=" ".join(a[1:])
```

```
path2=os.getcwd() + f"/Images_Attendance/"

path2=path2 + name + ".jpg"

pname=name.upper()

deleted.append(pname)

print("Printing Deleted List: ",deleted)

os.remove(path2)    #s.remove() method in Python is used to remove or delete a file path


data = pd.read_csv('User.csv')

print("User Before: ",data)

new_data = data[data.Name != name] #Create new list which not contain selected user

print("User After", new_data)

new_data.set_index('Name',inplace = True) ## setting Name as index column

print("New Dataset : ",new_data)

new_data.to_csv('User.csv') #save new data to User.csv file


for widget in frame.winfo_children():    #clear frame

    widget.destroy()


user_list(frame)    #show frame with updated list
```

```
def user_list(container):    #show user list and perform delete operation if require

    label = tk.Label(container, text ="List of Existing Users", font = "Helvetica", foreground="#263942")
```

```

label.config(font=("Helvetica", 15))

label.place(x = 140,y = 20)

#names = []

Lb1 = tk.Listbox(container,selectbackground = "lightblue",yscrollcommand = True,bg = "#ccc")    #change background
colour for selected user

```

```

data = pd.read_csv('User.csv') #get user list from user.csv

names = list(data.Name)

print("User present in User.csv file")

print(names)

for i in range(len(names)):

    Lb1.insert(i+1, f"{i+1}. {names[i]}")

Lb1.place(x = 90,y = 80)

ttk.Style().configure("TButton", padding=6, relief="flat",

    background="#ccc",foreground='green')

def back_clear_frame(frame):

    for widget in frame.winfo_children():

        widget.destroy()

admin(frame)

```

```

button1 = ttk.Button(container, text ="Delete",

    command = lambda : delete_selected(container,Lb1)) #call delete_seletected function with
given conatainer

button1.place(x = 300, y = 120)

```

```

button1 = ttk.Button(container, text ="Back",

    command = lambda : back_clear_frame(container)) #call back_clear_frame function

button1.place(x = 300, y = 180)

```

```

def sendNotification(personName,personImg):

    # Get the instance using access token

```

```

print("IN notification fun")

pb = PushBullet(access_token)

# Send the data by passing the main title

# and text to be send

time_now = datetime.datetime.now()

push1 = pb.push_note("WHO IS THERE",personName+": "+str(time_now.date()) + "-" + str(time_now.hour) + "-"
+str(time_now.minute) + "-" +str(time_now.second))

with open(personImg, "rb") as pic:

    file_data = pb.upload_file(pic, personName+": "+str(time_now.date()) + "-" + str(time_now.hour) + "-"
+str(time_now.minute) + "-" +str(time_now.second))

#file_data = self.pb.upload_file(imagedata, 'Motion detected: ' + personImg)

```

```

push = pb.push_file(**file_data)

# push2 = pb.push_file("Person",personImg, file_type="image/jpeg")

# Put a success message after sending

# the notification

print("Message sent successfully..")

```

```

def doorbell(button1,button2):

```

```

    button1['state'] = "disabled"

    button2['state'] = 'disabled'

    flg=0

    userName="UNKNOWN"

    userImg=None

    data = pd.read_csv("User.csv")

    names = list(data.Name)

    cap = cv2.VideoCapture(0)

    counter=0

    while True:

        success, img = cap.read()

        userImg=img

        if img is None:

```



```

        continue

    cv2.imshow('webcam', img)

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

    imgS = cv2.cvtColor(imgS, cv2.COLOR_BGR2RGB)

```

```

facesCurFrame = face_recognition.face_locations(imgS)

encodesCurFrame = face_recognition.face_encodings(imgS, facesCurFrame)

for encodeFace, faceLoc in zip(encodesCurFrame, facesCurFrame):

    matches = face_recognition.compare_faces(encodeListKnown, encodeFace)

    faceDis = face_recognition.face_distance(encodeListKnown, encodeFace)

    print("FACE ID:", faceDis)

    matchIndex = np.argmin(faceDis)

    if matches[matchIndex]:

        name = classNames[matchIndex].upper()

        print(name)

        r=0

        b=0

        g=255

        if(name in deleted):

            name="UNKNOWN"

            r=0

            b=255

            g=0

        print("Element found in deleted listt-----")

```

```

y1, x2, y2, x1 = faceLoc

y1, x2, y2, x1 = y1*4, x2*4, y2*4, x1*4

cv2.rectangle(img, (x1, y1), (x2, y2), (r, g, b), 2)

# cv2.rectangle(img, (x1, y2-35), (x2, y2), (r, g, b), cv2.FILLED)

# cv2.putText(img, name, (x1+6, y2-6), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255, 255), 2)

userName=name

```

```
flg=1
```

```
print("After flg change")
```

```
cv2.waitKey(2000)

cap.release()

cv2.destroyAllWindows()

print("Before break")

break

else:

    name = "UNKNOWN"

    print(name)

    y1, x2, y2, x1 = faceLoc

    y1, x2, y2, x1 = y1*4, x2*4, y2*4, x1*4

    cv2.rectangle(img, (x1, y1), (x2, y2), (0, 0, 255), 2)

    # cv2.rectangle(img, (x1, y2-35), (x2, y2), (0, 0, 255), cv2.FILLED)

    # cv2.putText(img, name, (x1+6, y2-6), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255, 255), 2)

    if(counter>=3):

        cv2.waitKey(2000)

        cap.release()

        cv2.destroyAllWindows()

        flg=1

        break

    counter=counter+1
```

```
if(flg==1 or counter>=3):

    break

# cv2.imshow('webcam', img)

if cv2.waitKey(20) & 0xFF == ord('q'):

    break

time_now = datetime.datetime.now()
```

```

    path1 = os.getcwd() + f"/results/" #{name}{time_now}.jpg"

    #print(frame)

    #print(path)

    s = path1+str(userName) + str(time_now.date()) + "-" + str(time_now.hour) + "-" +str(time_now.minute) + "-"
+str(time_now.second)

    cv2.imwrite(s+".jpg", userImg)

    # print("PATH: ",s)

    print("Before calling notification")

    sendNotification(userName,s+".jpg")

    button2['state']="normal"

    button1['state']="normal"

    # cap.release()

    # cv2.destroyAllWindows()

```

```

app = tk.Tk()    #creating application main window

app.title("Who is there?")

app.geometry("450x350")

app.resizable(False,False)

container = tk.Frame(app)    #It can be defined as a container to which, another widget can be added and organized

container.pack(side = "top", fill = "both", expand = True) #The Pack geometry manager packs widgets relative to the
earlier widget.

container.grid_rowconfigure(0, weight = 1)

container.grid_columnconfigure(0, weight = 1)

startpage(container)    #call statrpage function

```

```

def close():

    app.destroy()

```

```

def disable_event():

    pass

app.mainloop()

```

15.References:

1. W3school Webiste: <https://www.w3schools.com/python/>
2. Arsfutura – Luka Dulcic <https://arsfutura.com/magazine/face-recognition-with-facenet-and-mtcnn/> - 2020
3. Tutorials Point Website: <https://www.tutorialspoint.com/python/>
4. Face_regonition Doc aviliable on Github:
https://pythonhosted.org/face_recognition/readme.html
5. OpenCV official documentation:
https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html -2016
6. Willberger. (Deep learning Haar – cascade explained.) Internet:
www.willberger.org/cascade-haar-explained, Jan 13, 2018
7. Adrian Rosebrock . (Local binary patterns with python & OpenCV) internet:
<https://www.pyimagesearch.com/2015/12/07/local-binary--with-python-opencv/>
8. Pandas ([pandas documentation — pandas 1.4.3 documentation \(pydata.org\)](https://pandas.pydata.org/pandas-docs/stable/10min.html))
9. CSV ([csv — CSV File Reading and Writing — Python 3.10.5 documentation](https://docs.python.org/3.10.5/library/csv.html)).
10. Tkinter([tkinter — Python interface to Tcl/Tk — Python 3.10.5 documentation](https://docs.python.org/3.10.5/library/tkinter.html))
11. Agile ([Agile/Lean Documentation: Strategies for Agile Software Development \(agilemodeling.com\)](https://agilemodeling.com/))

