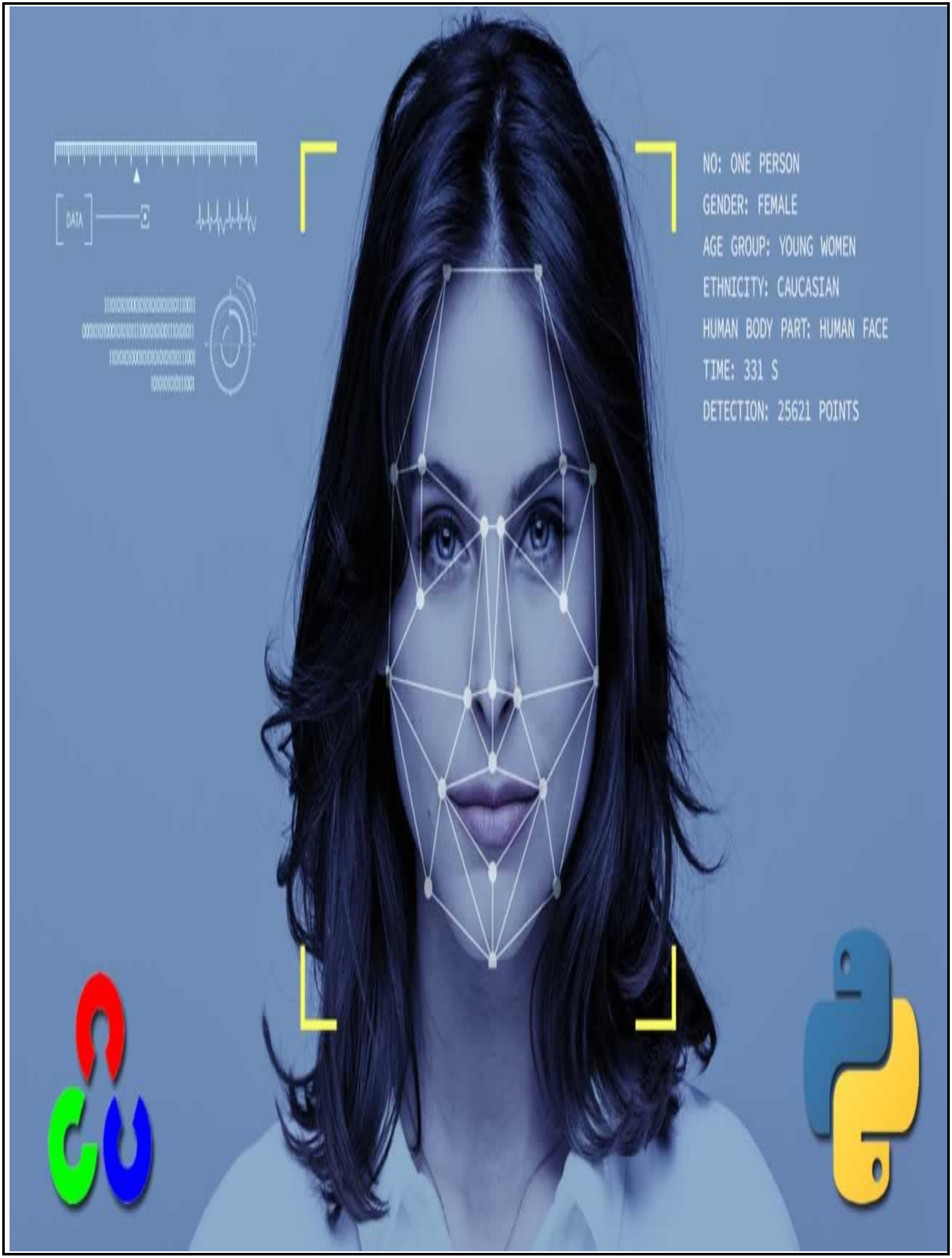

DATA
NO: ONE PERSON
GENDER: FEMALE
AGE GROUP: YOUNG WOMEN
ETHNICITY: CAUCASIAN
HUMAN BODY PART: HUMAN FACE
TIME: 331 S
DETECTION: 25621 POINTS

A

Project Report

On

Face Recognition Attendance System Using Python

Submitted in partial fulfilment of the requirements for the award of a degree

Bachelor of Computer Application

From

Hemchand Yadav Vishwavidyalaya, Durg (C.G.)

Year: 2022-2023

Guide: Submitted by:

Mrs. Rupali Kharche Simarjit Singh

Roll No: 33350130022

Submitted to

SWAMI SHRI SWAROOPANAND SARASWATI MAHAVIDYALAYA HUDCO, BHILAI

Hemchand Yadav Vishwavidyalaya, Durg (C.G.)

Certificate of Approval

This is to certify that the Project work entitled "***Face Recognition Attendance System Using Python***" is carried out by Mr. SIMARJIT SINGH. A student of BCA-III at **SWAMI SHRI SWAROOPANAND SARASWATI MAHAVIDYALAYA HUDCO, BHILAI** is hereby approved as a credible work in the discipline of Computer Science and Application for the award of the degree of **Bachelor of Computer Application** during the year 2022-2023 from **Hemchand Yadav Vishwavidyalaya, Durg (C.G.)**

Mrs. Rupali Kharche

H.O.D CS Dept,

CERTIFICATE

This is to certify that the Project work entitled

“Face Recognition Attendance System Using Python”

Submitted to the **SWAMI SHRI SWAROOPANAND SARASWATI MAHAVIDYALAYA HUDCO, BHILAI** by **Mr Simarjit Singh** Roll No 33350130022, in partial fulfilment of the requirements relating to the nature and standard of the award of Bachelor of Computer Application degree by, Hemchand Yadav Vishwavidyalaya, Durg (C.G.) for the academic year 2022 -2023. This project work has been carried out under my guidance.

Mrs. Rupali Kharche

H.O.D CS Dept.

____

CERTIFICATE OF THE ORGANIZATION

This is to certify that **Mr. Simarjit Singh** Student of **BCA–III** of "SWAMI SHRI SWAROOPANAND SARASWATI MAHAVIDYALAYA HUDCO, BHILAI" has completed his project work titled "**Face Recognition Attendance System Using Python**" as part of his course curriculum.

He has done his project using Python Tkinter and OpenCV during the period from, October 2022 to January 2023. He has completed the project well within the time frame. He is sincere and, hardworking and his conduct during the project is commendable.

ORGANIZATION NAME

An ISO 9001:2015 Certified Company
SEED IT SOLUTION
Learn By IT Professionals

AUTHORISED SIGNATURE ______________________________

CERTIFICATE OF EVALUATION

This is to certify that the Project work entitled "**Face Recognition Attendance System Using Python**" is carried out by Mr. **Simarjit Singh**, a student of BCA-III year at **SWAMI SHRI SWAROOPANAND SARASWATI MAHAVIDYALAYA HUDCO, BHILAI**, after proper evaluation and examination, is hereby approved as a credible work in the discipline of Computer Science and Application and is done satisfactorily for its acceptance as a requisite for the award of the degree of Bachelor of Computer Application during the year 2022-2023 from Hemchand Yadav Vishwavidyalaya, Durg (C.G.).

Internal Examiner

____

External Examiner

____

DECLARATION

This is to certify that the Project work entitled "**Face Recognition Attendance System Using Python**", which is submitted by me in partial fulfilment for the award of the degree of Bachelor of Computer Application, **SWAMI SHRI SWAROOPANAND SARASWATI MAHAVIDYALAYA HUDCO, BHILAI**, comprises the original work carried out by me. I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full for the award of any other degree or diploma in this Institute or any other Institute or University.

Place: BHILAI

Date:

SIMARJIT SINGH

Roll No: 33350130016

ACKNOWLEDGEMENT

We are very delighted with the accomplishment of the project in Python, which was very educational and practically beneficial. It is due to his encouragement and persistent motivation that I could extend the scope of the project to a much more useful data processing report. We have especially thankful to the "**SWAMI SHRI SWAROOPANAND SARASWATI MAHAVIDYALAYA HUDCO, BHILAI"** who helps us with his deep knowledge and practical experience in computer science, rendered all possible in fulfilment of the project and for explaining the method of approach. So, system development is the pooling of talents, extending help and cooperation above as a team effort.

I take this opportunity to express my deep sense of gratitude and wholehearted thanks to my project guide **"Mrs. Rupali Kharche"** for their valuable guidance, keep interest and affection encouragement through the work and correcting various documents of mine with attention and care.

Table OF Contents

INTRODUCTION

Abstract:

This report presents a comprehensive overview of the Face Recognition project using Python and OpenCV. The project aims to develop a software solution for recognizing faces in real-time video feeds and images. The report covers the various stages of the project, including the design, development, testing, and deployment of the solution.

Intro:

Face recognition is a biometric technique that involves the use of computer algorithms to identify or verify the identity of individuals based on their facial features. This technology has a wide range of applications, including security systems, access control, and identification systems. In recent years, the advancement of computer vision and deep learning algorithms has led to the development of high-accuracy face recognition systems.

The purpose of this major report is to present a comprehensive study on face recognition technology and its implementation using Python and OpenCV. The report will cover the key concepts, techniques, and algorithms used in face recognition, as well as provide an in-depth analysis of the development and implementation of a face recognition system. This report will also provide insights into the challenges faced during the development process and discuss future directions for this technology. The the report concludes with a discussion of the significance and impact of face recognition technology in various domains and industries.

Methodology:

The methodology for a face recognition project using Python and OpenCV can be outlined as follows:

1. Collect and pre-process data: Gather a dataset of face images and pre-process the images to ensure that they are in a consistent format and size.
2. Train the model: Use a deep learning algorithm, such as Convolutional Neural Networks (CNNs), to train the model on the pre-processed dataset. The model learns to recognize faces by comparing the patterns and features in the images.
3. Evaluate the model: Evaluate the performance of the model using various metrics, such as accuracy, precision, recall, and F1-score.
4. Deploy the model: Deploy the trained model on a suitable platform, such as a desktop, laptop, or mobile device, to perform face recognition in real time.
5. Improve the model: Continuously improve the model by tweaking its parameters and architecture, adding more data to the dataset, or using advanced deep learning techniques, such as transfer learning or fine-tuning.

Note: The exact methodology may vary depending on the specific requirements of the project and the preferences of the developer. However, the above steps provide a general overview of the process of building a face recognition system using Python and OpenCV.

Background:

The concept of face recognition has been widely researched and studied for decades. With the rise of technology and increased computational power, the field of face recognition has experienced rapid growth and advancement. Face recognition refers to the process of identifying a person based on their facial features, such as eyes, nose, mouth, and other unique characteristics. It is an important aspect of computer vision and is widely used in various applications, such as security systems, access control systems, and biometric identification.

The technology behind face recognition has improved dramatically in recent years, and is now becoming increasingly popular. OpenCV, a computer vision library, provides several algorithms for

face recognition that is efficient and accurate. Python, being
A high-level programming language is an ideal choice for developing
face recognition systems due to their simplicity and ease of use.

This major report will provide a comprehensive overview of face recognition using Python and OpenCV. It will cover the background of the technology, the methodology used, and the results obtained. The report will also provide insights into the future of face recognition and its potential applications.

Objective:

The objectives of the face recognition project can include the following:

1. Developing a system that can accurately identify individuals based on their facial features.
2. Improving the speed and efficiency of the recognition process by using advanced algorithms and techniques.
3. Implementing the system in a way that makes it user-friendly and easy to use.
4. Ensuring the security and privacy of the user’s personal information.
5. Investigating and implementing new technologies and techniques that can further improve the accuracy and speed of the recognition process.
6. Evaluating the performance of the system in different conditions and environments.
7. Identifying and addressing any limitations or challenges faced during the development process.
8. Providing recommendations for future improvements to the system

Purpose, Scalability and Applicability:

The purpose of the Face Recognition project is to develop a system that can accurately recognize and identify individual faces from a database. The scope of the project involves the use of Python programming language and OpenCV library to develop the face recognition system.

The applicability of the project is in various fields such as security systems, access control, and identity verification. The face recognition system can be used to grant access to secure areas based on the identification of the individual's face. It can also be used in identity verification processes in various organizations, helping to streamline and automate the process of verifying the identity of individuals. The project has the potential to impact various industries including banking, government, and retail.

In summary, the purpose, scope, and applicability of the Face Recognition project are to develop a system that can accurately recognize and identify individual faces for various security and identity verification purposes.

Survey of Technology

The survey of technology for face recognition involves reviewing the existing technologies and methods used for face detection and recognition. This includes an overview of traditional methods such as eigenface, fisher face, and local binary patterns (LBP), as well as more recent deep learning-based methods. The survey also covers popular face recognition datasets, evaluation metrics, and software tools and libraries commonly used for face recognition.

Additionally, the survey should also cover the current state-of-the-art face recognition algorithms and the challenges and limitations faced by existing systems. This would provide a comprehensive overview of the field of face recognition and give an understanding of the various factors that have shaped its development over time.

It is important to note that the survey on technology should be updated regularly to reflect the latest advancements and innovations in face recognition technology.

Technology Used

In the face recognition project using Python and OpenCV, the following technology is used:

1. **Python:** Python is a high-level, interpreted and dynamically typed language that is widely used for building various applications. It has a large community and a rich library which makes it a popular choice for computer vision applications.
2. **OpenCV:** OpenCV (Open-Source Computer Vision Library) is a free and open-source library for computer vision programming. It contains over 2500 computer vision algorithms and supports C++, Python, and Java. OpenCV provides a comprehensive suite of computer vision algorithms including face detection and face recognition.
3. **Haarcascade:** Haarcascade is a machine learning-based object detection algorithm that is used to identify objects in images and videos. They are commonly used in computer vision applications to detect faces, eyes, etc. In the face recognition project, Haar cascades are used to detect faces in an image.
4. **Local Binary Patterns Histograms (LBPH):** LBPH is a texture-based algorithm used for face recognition. It measures the local structure of the face and converts it into a histogram representation which is then used for comparison.
5. **NumPy:** NumPy is a library for the Python programming language that provides support for large multi-dimensional arrays and matrices. It is used in the project for data manipulation and array operations.
6. **Matplotlib:** Matplotlib is a plotting library for Python that is used for data visualization. In the project, it is used to display the captured images and the detected faces.

Requirement and Analysis

Problem Definition

The problem definition in a face recognition project can refer to a variety of challenges or limitations that may arise during the development and implementation of the system. Some common problems that might be encountered include:

1. Accuracy: The ability of the system to accurately detect and identify faces in real-world conditions.

2. Scalability: The ability of the system to handle increasing numbers of users or images, and maintain its performance.

3. Privacy: Ensuring the protection of personal information, such as facial images and biometric data.

4. Lighting conditions: The ability of the system to perform well in varying lighting conditions, including low-light environments and bright sunlight.

5. Occlusions: The ability of the system to handle cases where a person's face is partially obscured by hair, glasses, or other objects.

6. Diversity: The ability of the system to handle differences in facial features across different cultures and demographic groups.

7. False negatives and false positives: The ability of the system to minimize instances of incorrectly identifying a person or failing to recognize a person's face.

Requirement Specifications

1. The requirement specification is a crucial aspect of software development as it outlines the requirements and functionalities needed in the final product. In the case of a Face Recognition project, the following requirements should be specified:

2. Image Acquisition: The system should be able to acquire images from various sources such as a webcam, digital camera, or from an existing dataset.

3. Image Processing: The system should be able to process images to extract and manipulate facial features for accurate recognition.

4. Feature Extraction: The system should extract relevant features from the images, such as eyes, nose, mouth, and cheekbones, to identify a face.

5. Recognition Algorithm: The system should have a robust recognition algorithm to match the extracted features with the existing database of faces.

6. Performance Metrics: The system should have performance metrics to measure the accuracy and efficiency of the recognition process.

7. User Interface: The system should have an intuitive and user-friendly interface to allow users to interact with the system and view the results.

8. Security and Privacy: The system should have measures in place to protect the privacy and security of personal data and images stored in the system.

These requirements should be specified and agreed upon by the stakeholders to ensure that the final product meets their expectations and requirements.

Software and Hardware Requirements

Software requirements for a face recognition project using Python and OpenCV include:

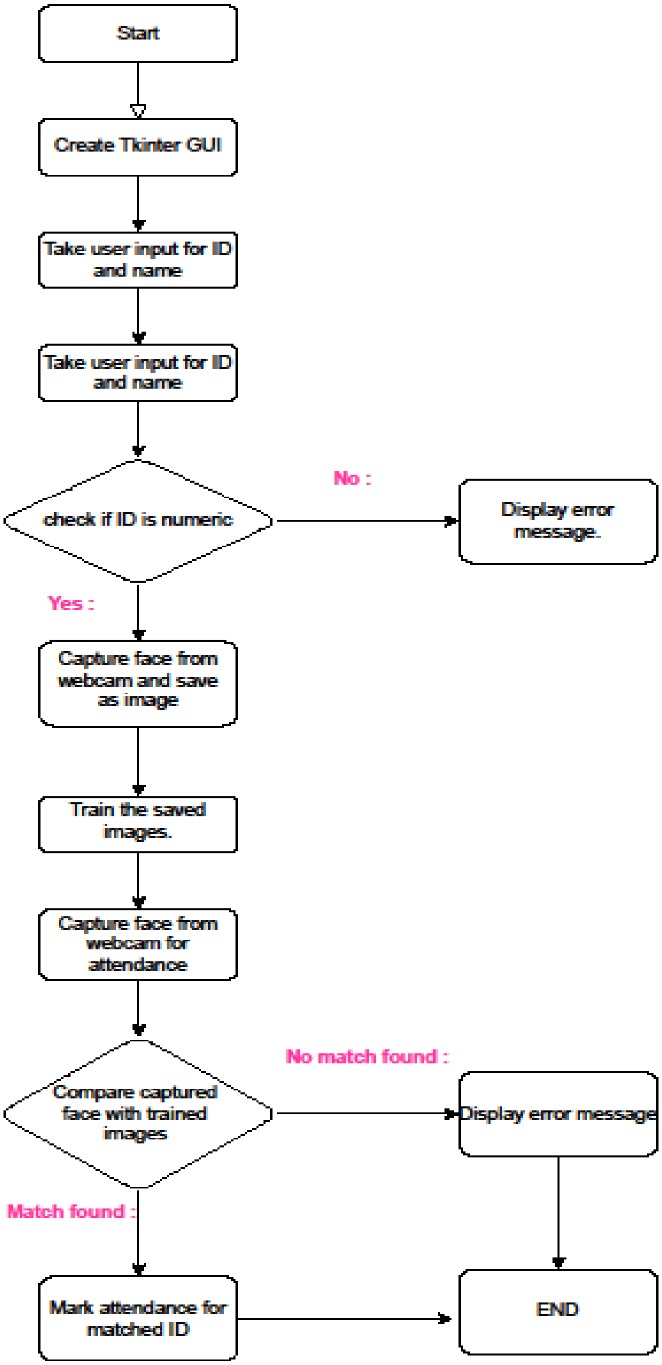
- Python 3.11.1
- OpenCV library for Python (can be installed using pip install OpenCV-python)
- NumPy library for Python (can be installed using pip install NumPy)
- Utils library (can be installed using pip install imutils)
- Matplotlib library for Python (can be installed using pip install matplotlib)

Hardware requirements depend on the complexity and scale of the project but generally include:

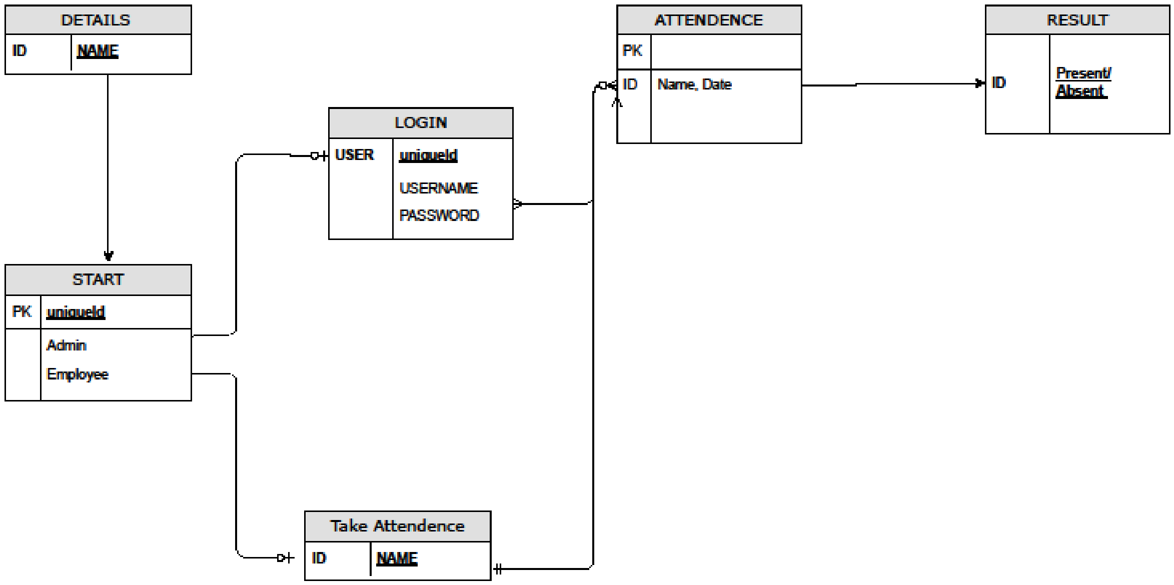
- A computer with a processor capable of running Python and OpenCV efficiently (at least Intel Core i3 or higher)
- At least 4GB of RAM
- At least 128GB of SSD (for smooth working)
- A webcam or other image-capture device

System Design

Flow Chart


Start
Create Tkinter GUI
Take user input for ID and name
Take user input for ID and name
check if ID is numeric
No :
Display error message.
Yes :
Capture face from webcam and save as image
Train the saved images.
Capture face from webcam for attendance
Compare captured face with trained images
No match found :
Display error message
Match found :
Mark attendance for matched ID
END

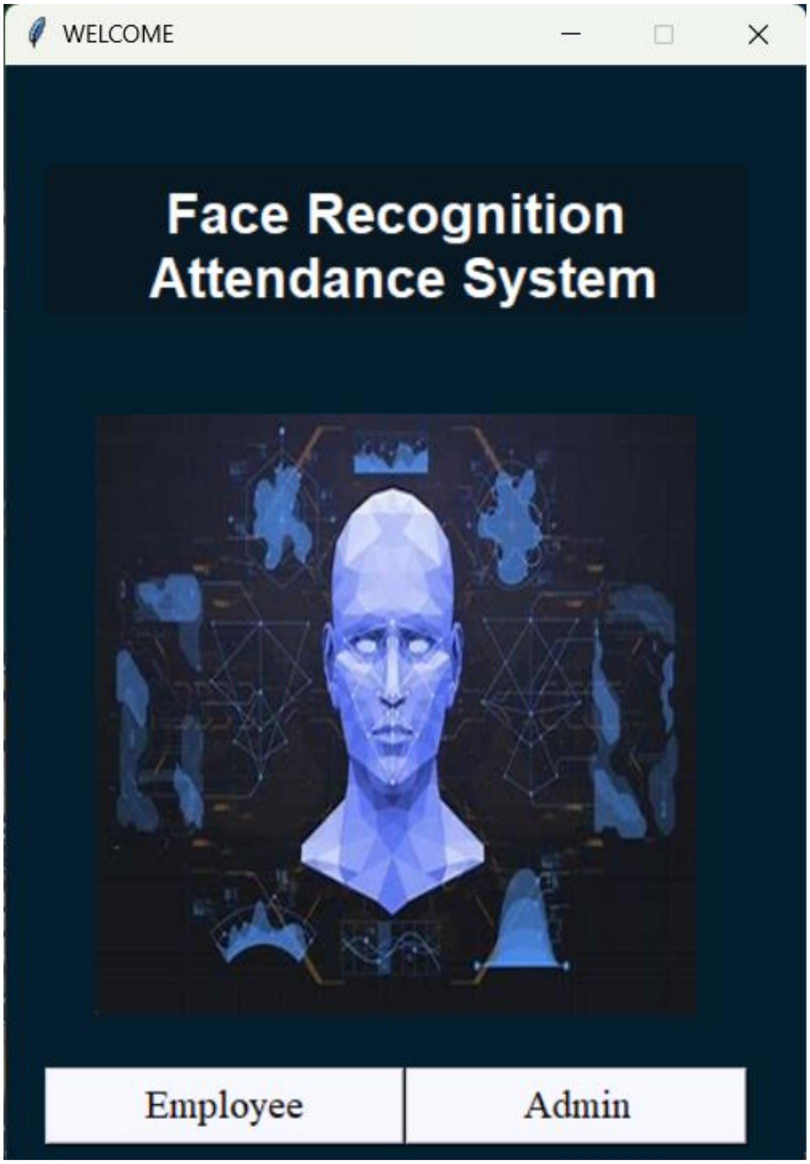
ER Diagram


DETAILS
ID
NAME
LOGIN
USER
uniqueId
USERNAME
PASSWORD
ATTENDENCE
PK
ID
Name, Date
RESULT
ID
Present/
Absent
START
PK
uniqueId
Admin
Employee
Take Attendence
ID
NAME

SOURCE CODE

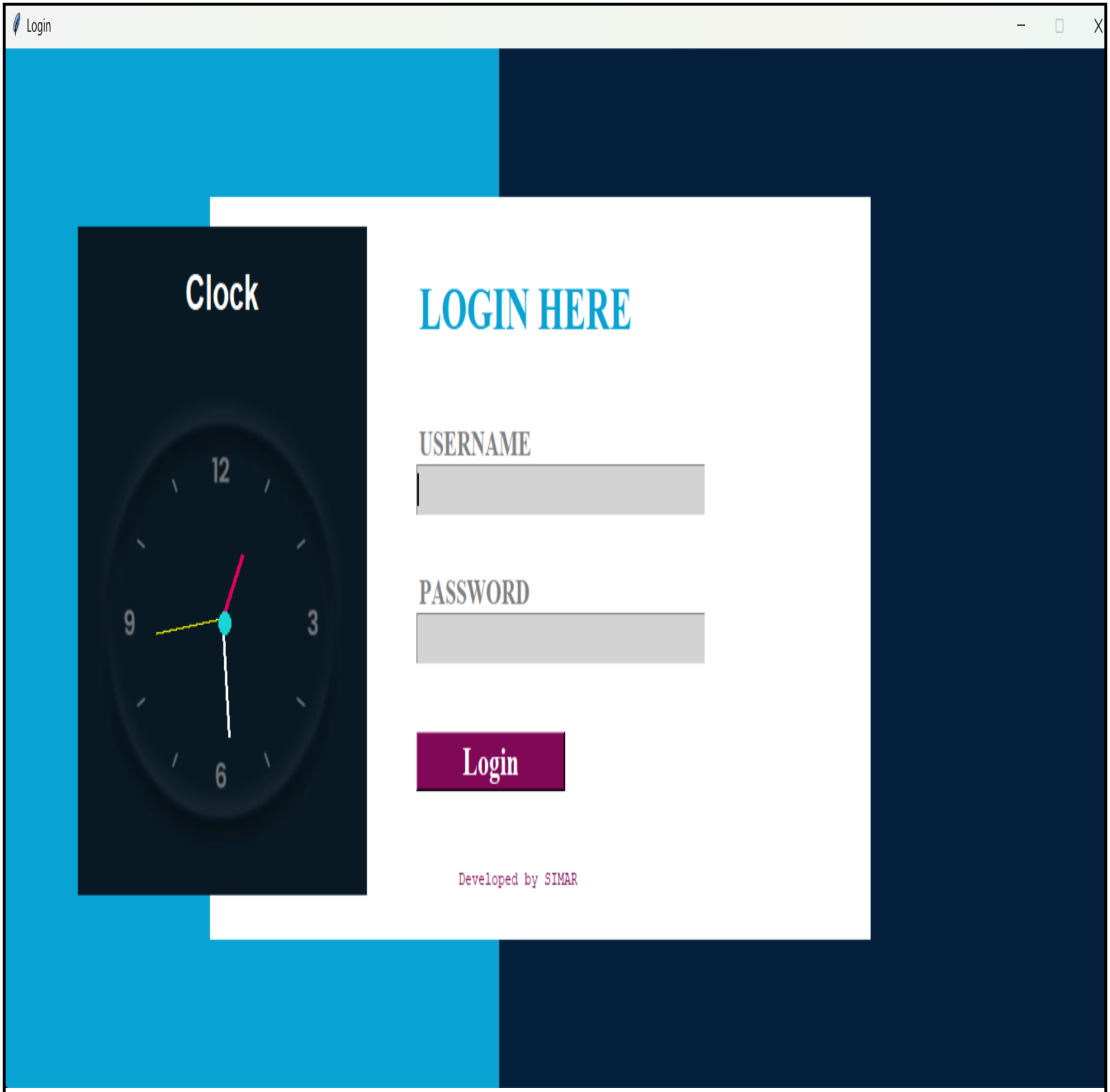
```
from tkinter import *
from tkinter import messagebox
from attendence import Attendance
from PIL import Image, ImageTk, ImageDraw
from math import *
from datetime import *


class Main:
    def __init__(self, root):
        self.password_text = None
        self.username_text = None
        self.username = None
        self.frame = None
        self.heading = None
        self.code = None
        self.new_login_button = None
        self.img_label = None
        self.img = None
        self.login_frame = None
        self.login_button = None
        self.register_button = None
        self.password_entry = None
        self.new_employee = None
        self.img = None
        self.origin = None
        self.clock = None
        self.draw = None
        self.bg = None
        self.h = None
        self.m = None
        self.s = None
        self.hr = None
        self.min_ = None
        self.sec_ = None

        self.root = root
        self.root.geometry('400x400+0+0')
        self.root.title("WELCOME")
        self.root.config(bg="#021e2f")
        self.root.resizable(bool(0), bool(0))

        # Buttons
        self.employee = Button(self.root, text='Employee', font=('Times New Roman', 15), bg='ghostwhite', width=10
                               , command=self.employee)
        self.employee.place(x=30, y=80)
        self.admin = Button(self.root, text='Admin', font=('Times New Roman', 15), bg='ghostwhite', width=10,
                            command=self.admin)
        self.admin.place(x=200, y=80)

        self.copyWrite = Text(self.root, foreground='khaki',
background=self.root.cget("background"), borderwidth=0)
        self.copyWrite.tag_configure("superscript", offset=4)
        self.copyWrite.insert("insert", "Developed by SIMAR")
        self.copyWrite.configure(state="disabled")
        self.copyWrite.pack(side="top")
        self.copyWrite.place(x=570, y=300)

    def employee(self):
        self.new_employee = Toplevel()
        self.new_employee.focus()
        Attendance(self.new_employee)

    def admin(self):
        def login():
            def search():
                self.f =
open("F:/Face_Recognition_Attendence_System/venv/att.csv", "r")
                c = 0
                b = []
                for i in range(1, 31):
                    b.append("Absent")

                for line in self.f:
                    self.x = line.split(" ")
                    if self.x[0] == self.id_entry.get():
                        self.l1 = Label(self.report, text=self.x[0] + "
" + self.x[1] + " " + self.x[2] + " ",
                                        bg="khaki")
                        m = self.x[2].split("-")
                        for i in range(0, 30):
                            if i == int(m[2]):
                                b[i] = "Pres"
                        self.l1.grid(row=c + 2, column=0)
                        c += 1
                c1 = 0
                c2 = 0
                self.t = Toplevel()
                for i in range(1, 30):
                    if b[i] == "Absent":
                        c1 = c1 + 1
                        self.l = Label(self.t, text=str(i) + " " +
str(b[i]) + " ", fg="red")
                    else:
                        c2 = c2 + 1
                        self.l = Label(self.t, text=str(i) + " " +
str(b[i]) + " ", fg="blue")

                    self.l.grid(row=i, column=0)
                # print(c1, c2)
            if self.username.get() == 'Admin' and 
self.password_entry.get() == 'Root':
                self.login_frame.focus()
                self.report = Toplevel()
                self.report.focus()
                self.report.config(bg="khaki")
                self.report.resizable(0, 0)

                self.id_label = Label(self.report, text="ID", width=10, 
bg="khaki")
                self.id_label.grid(row=0, column=0)
                self.id_entry = Entry(self.report)
                self.id_entry.focus()
                self.id_entry.grid(row=0, column=1)
                self.go_button = Button(self.report, text="Go", 
width=10, command=search, bg='ghostwhite')
                self.go_button.grid(row=0, column=2)
                self.login_frame.destroy()

            elif self.username.get() != 'Admin' and 
self.password_entry.get() != 'Root':
                messagebox.showerror("invalid", "invalid username and 
password")
            elif self.password_entry.get() != 'Root':
                messagebox.showerror("invalid", "invalid  password")
            elif self.username.get() != "Admin":
                messagebox.showerror("invalid", "invalid username ")

        def clock_image(hr, min_, sec_):
            self.clock = Image.new("RGB", (400, 400), (8, 25, 35))
            self.draw = ImageDraw.Draw(self.clock)
            # =====For Clock Image
            self.bg = 
Image.open("../Face_Recognition_Attendence_System/images/c.png")
            self.bg = self.bg.resize((300, 300), 
Image.Resampling.LANCZOS)
            self.clock.paste(self.bg, (50, 50))

            self.origin = 200, 200
            # ====Hour Line Image====
            self.draw.line((self.origin, 200 + 50 * sin(radians(hr)), 
200 - 50 * cos(radians(hr))), fill="#DF005E", 
                           width=4)
            # ====Min Line Image====
            self.draw.line((self.origin, 200 + 80 * sin(radians(min_)), 
200 - 80 * cos(radians(min_))), fill="white", 
                           width=3)
            # ====Sec Line Image====
            self.draw.line((self.origin, 200 + 80 * sin(radians(sec_)), 
200 - 100 * cos(radians(sec_))), fill="yellow", 
                           width=2)
            self.draw.ellipse((195, 195, 210, 210), fill="#1AD5D5")
            self.clock.save("clock.png")
        def working():
            self.h = datetime.now().time().hour
            self.m = datetime.now().time().minute
            self.s = datetime.now().time().second
            self.hr = (self.h / 12) * 360
            self.min_ = (self.m / 60) * 360
            self.sec_ = (self.s / 60) * 360
            clock_image(self.hr, self.min_, self.sec_)
            self.img = ImageTk.PhotoImage(file="clock.png")
            self.clock_label.config(image=self.img)
            self.clock_label.after(200, working)

        self.login_frame = Toplevel()
        self.login_frame.title("Login")
        self.login_frame.geometry("1350x700+0+0")
        self.login_frame.config(bg="#021e2f")
        self.login_frame.resizable(bool(0), bool(0))

        # ===Background Colors====================
        self.left_label = Label(self.login_frame, bg="#08A3D2", bd=0)
        self.left_label.place(x=0, y=0, relheight=1, width=600)

        self.right_label = Label(self.login_frame, bg="#031F3C", bd=0)
        self.right_label.place(x=600, y=0, relheight=1, relwidth=1)

        # ===Frames===================
        self.login_frame_1 = Frame(self.login_frame, bg="white")
        self.login_frame_1.place(x=250, y=100, width=800, height=500)

        self.title = Label(self.login_frame_1, text="LOGIN HERE",
font=("times new roman", 30, "bold"), bg="white",
                           fg="#08A3D2")
        self.title.place(x=250, y=50)

        self.username_text = Label(self.login_frame_1, text="USERNAME",
bg="white", fg="grey",
                                   font=("times new roman", 18, "bold"))
        self.username_text.place(x=250, y=150)

        self.username = Entry(self.login_frame_1, font=("times new
roman", 15), bg="lightgrey")
        self.username.place(x=250, y=180, width=350, height=35)
        self.username.focus()

        self.password_text = Label(self.login_frame_1, text="PASSWORD",
bg="white", fg="grey",
                                   font=("times new roman", 18, "bold"))
        self.password_text.place(x=250, y=250)

        self.password_entry = Entry(self.login_frame_1, font=("times new
roman", 15), bg="lightgrey", show="*")
        self.password_entry.place(x=250, y=280, width=350, height=35)

        self.new_login_button = Button(self.login_frame_1, text="Login",
font=("times new roman", 20, 'bold'),
                                       fg="white", bg="#800857",
command=login)
        self.new_login_button.place(x=250, y=360, width=180, height=40)

        # ===Clock===================
        self.clock_label = Label(self.login_frame, text="\nClock",
font=("Book Antigua", 25, 'bold'), fg="white",
                                 compound=BOTTOM, bg="#081923", bd=0)
        self.clock_label.place(x=90, y=120, height=450, width=350)

        working()

        self.copyWrite = Text(self.login_frame_1, foreground='#800857',
background='white',
                              borderwidth=0)
        self.copyWrite.tag_configure("superscript", offset=4)
        self.copyWrite.insert("insert", "Developed by SIMAR")
        self.copyWrite.configure(state="disabled")
        self.copyWrite.pack(side="top")
        self.copyWrite.place(x=300, y=450)


if __name__ == '__main__':
    Main(Tk())
    mainloop()
```

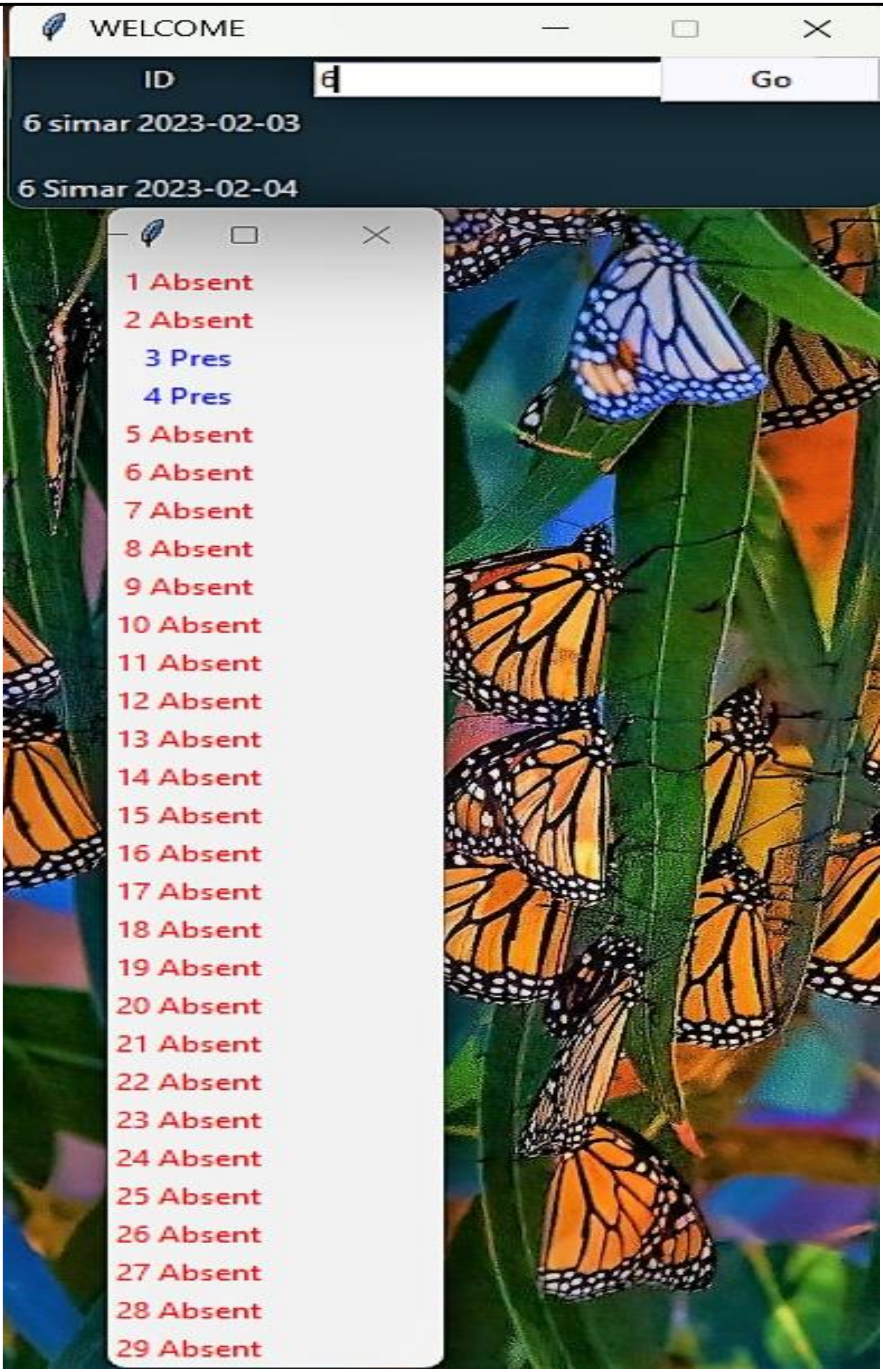
SCREENSHOT



WELCOME

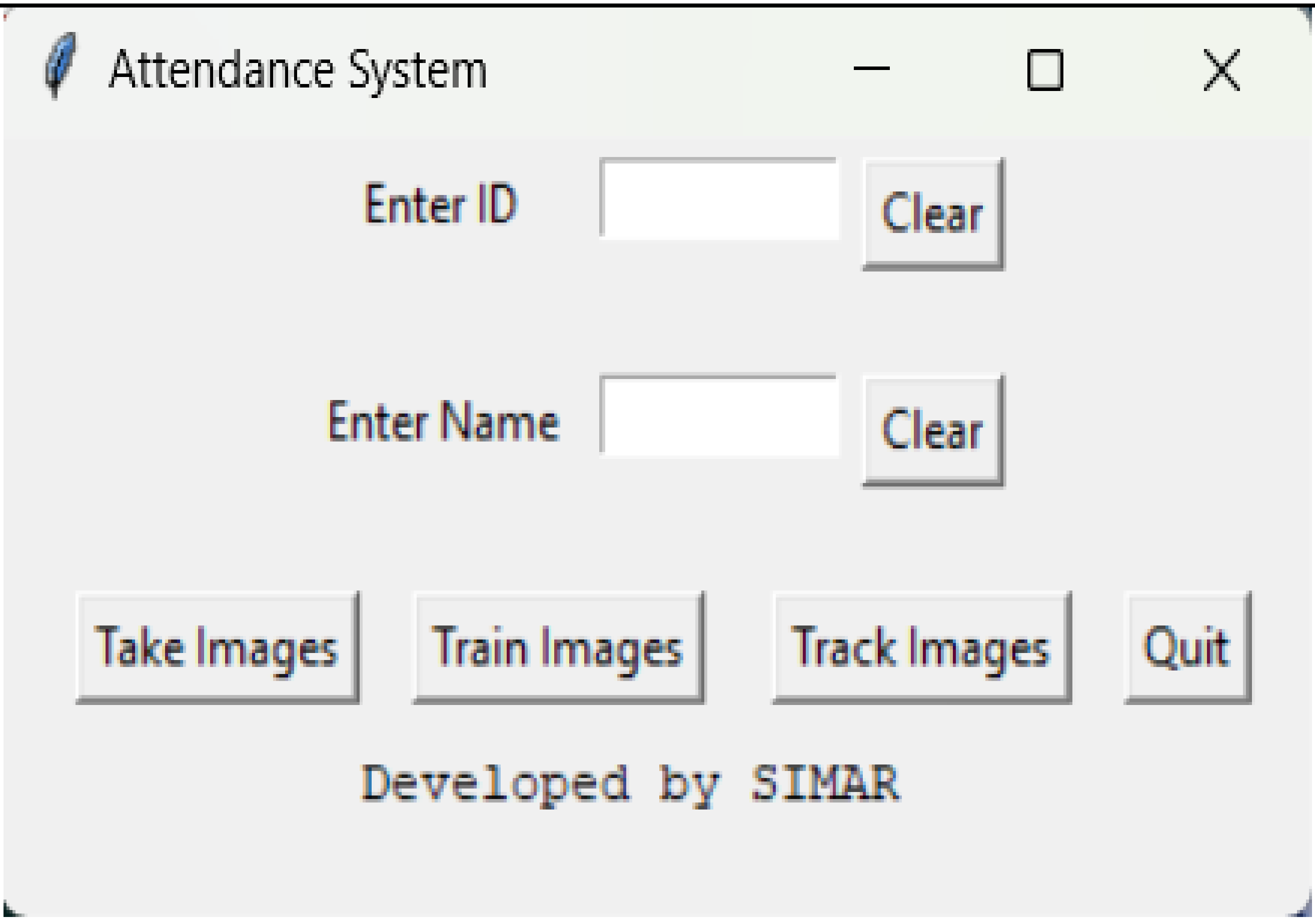
Face Recognition Attendance System

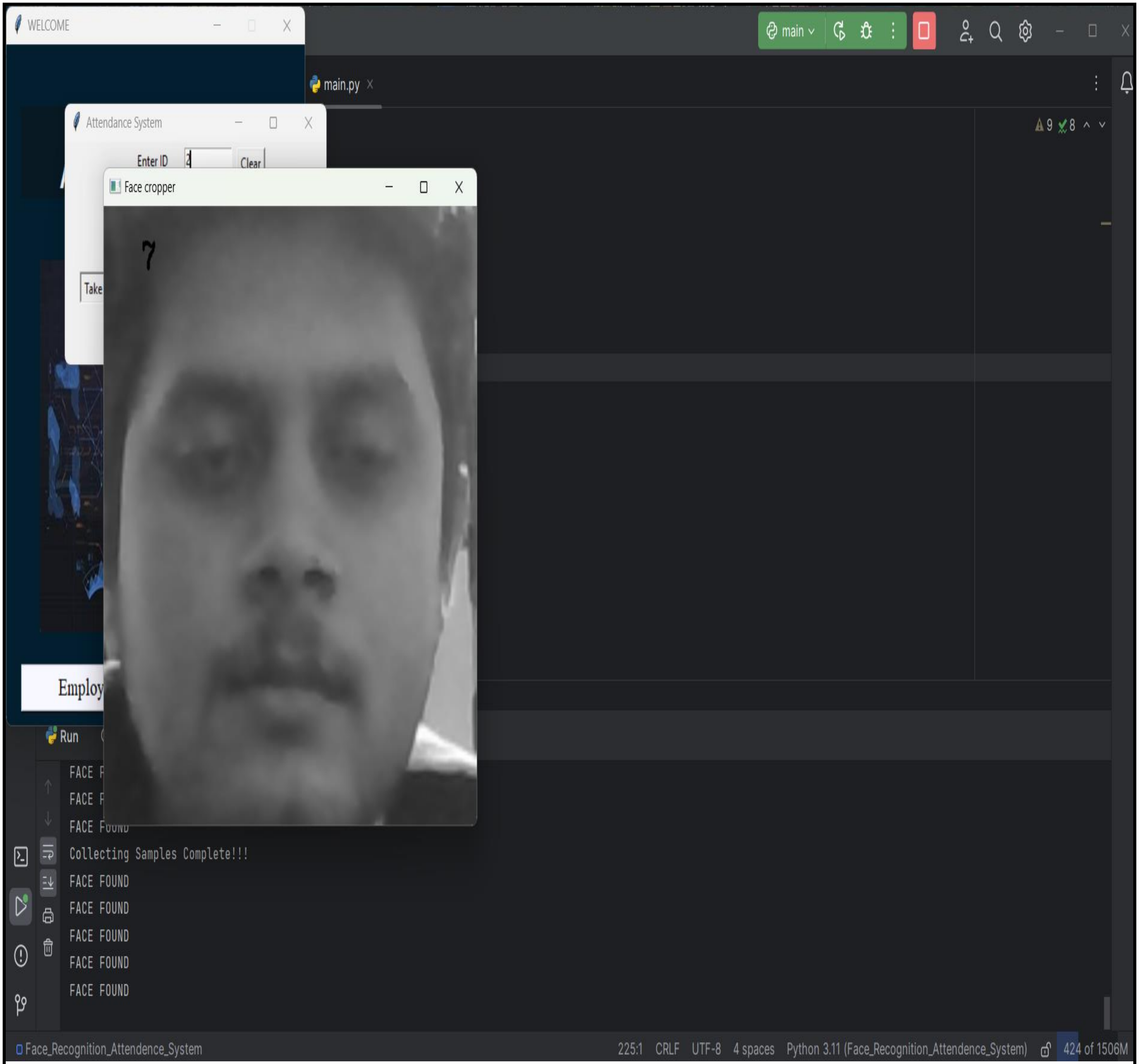

Login
Clock
12
9
3
6
LOGIN HERE
USERNAME
PASSWORD
Login
Developed by SIMAR



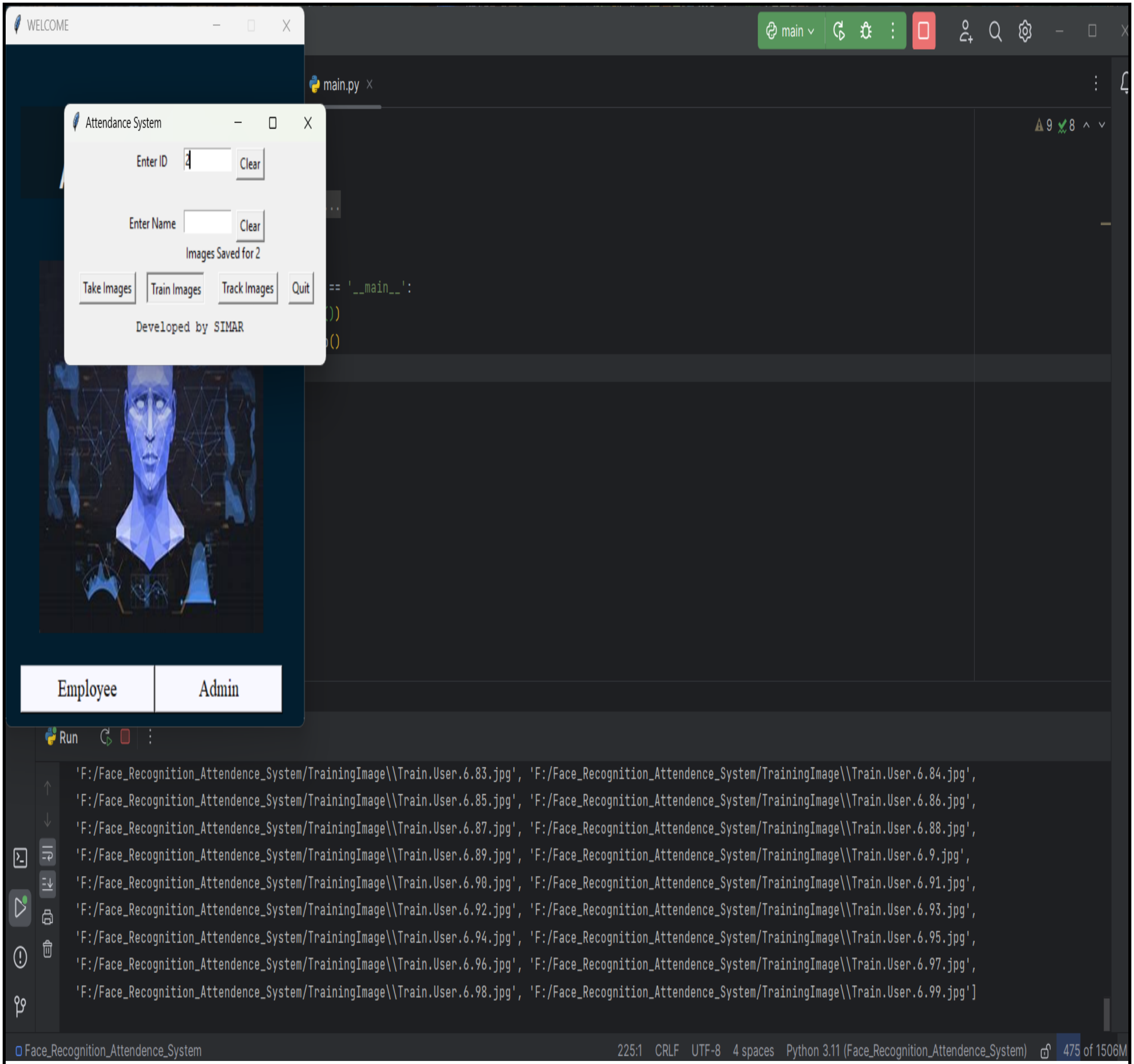
```
ID  6
6 simar 2023-02-03

6 Simar 2023-02-04
```

```
1 Absent
2 Absent
3 Pres
4 Pres
5 Absent
6 Absent
7 Absent
8 Absent
9 Absent
10 Absent
11 Absent
12 Absent
13 Absent
14 Absent
15 Absent
16 Absent
17 Absent
18 Absent
19 Absent
20 Absent
21 Absent
22 Absent
23 Absent
24 Absent
25 Absent
26 Absent
27 Absent
28 Absent
29 Absent
```


Attendance System
Enter ID
Clear
Enter Name
Clear
Take Images
Train Images
Track Images
Quit
Developed by SIMAR



```
FACE F
FACE F
FACE FOUND
Collecting Samples Complete!!!
FACE FOUND
FACE FOUND
FACE FOUND
FACE FOUND
FACE FOUND
```



Enter ID 2

Enter Name

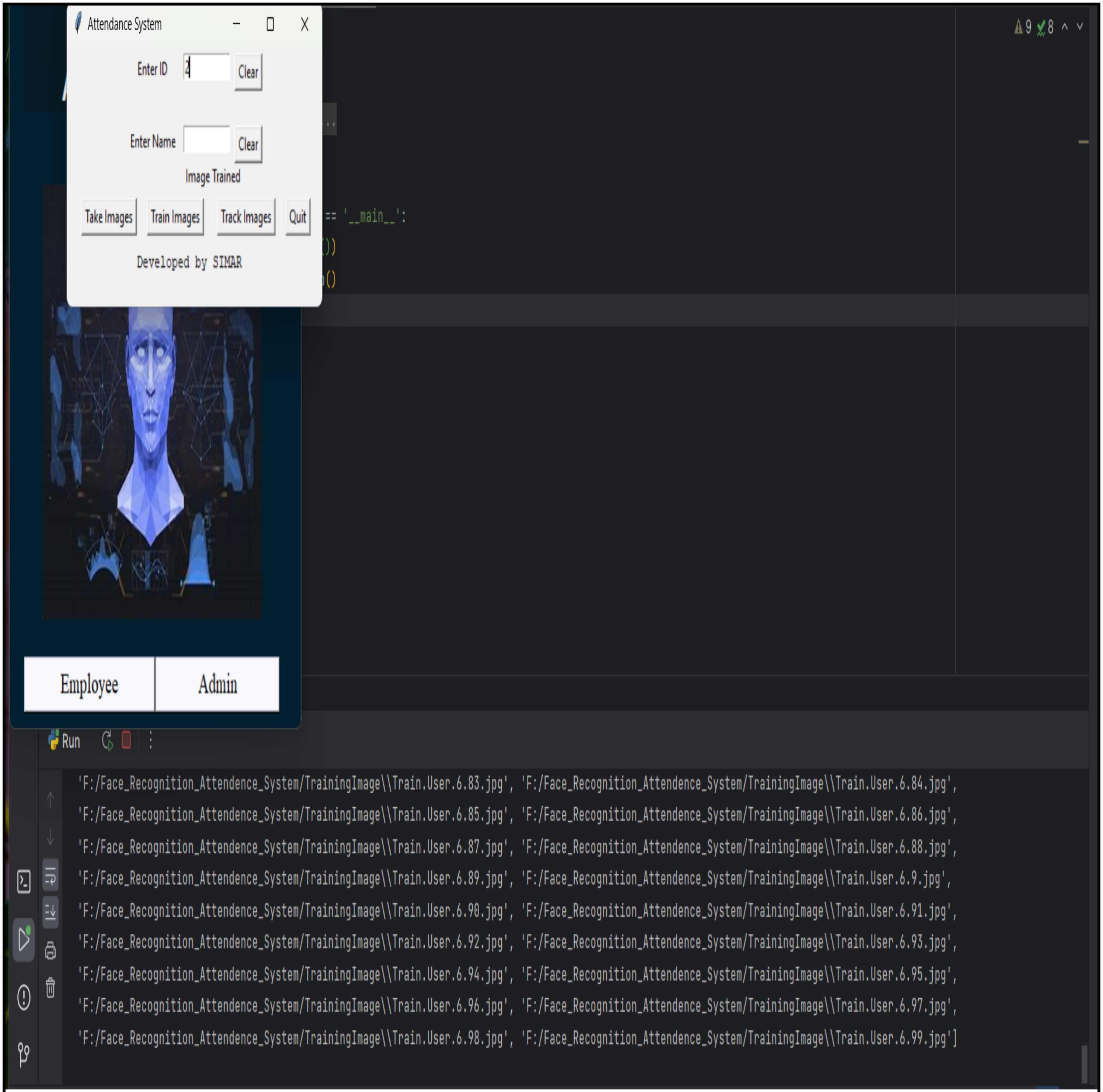
Images Saved for 2

Developed by SIMAR

```
== '__main__':
())
()
```

```
'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.83.jpg', 'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.84.jpg',
'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.85.jpg', 'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.86.jpg',
'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.87.jpg', 'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.88.jpg',
'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.89.jpg', 'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.9.jpg',
'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.90.jpg', 'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.91.jpg',
'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.92.jpg', 'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.93.jpg',
'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.94.jpg', 'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.95.jpg',
'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.96.jpg', 'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.97.jpg',
'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.98.jpg', 'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.99.jpg']
```

Face_Recognition_Attendence_System   225:1   CRLF   UTF-8   4 spaces   Python 3.11 (Face_Recognition_Attendence_System)   475 of 1506M



Enter ID 2

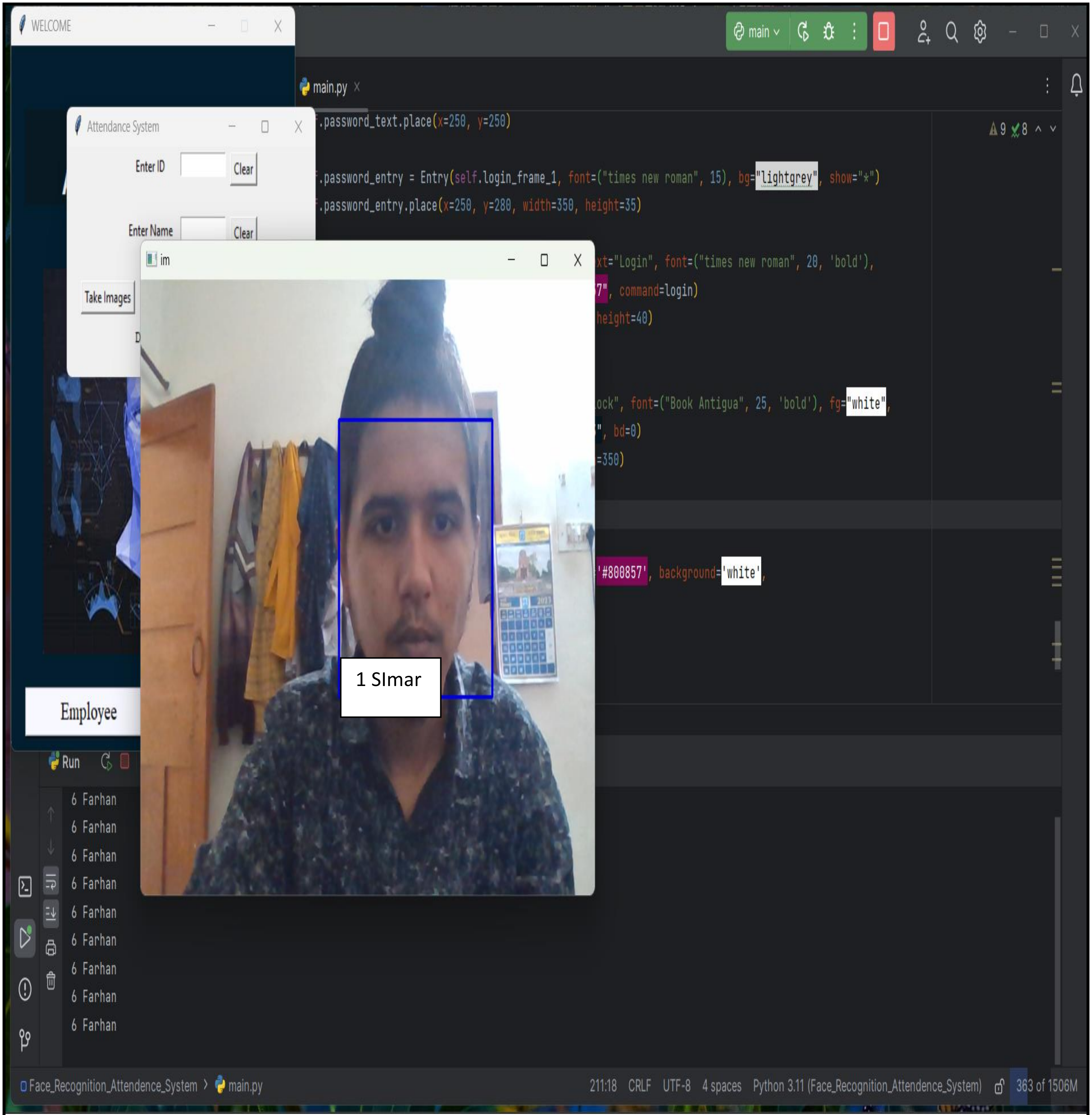
Enter Name

Image Trained

Developed by SIMAR

```
== '__main__':
))
()
```

```
'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.83.jpg', 'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.84.jpg',
'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.85.jpg', 'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.86.jpg',
'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.87.jpg', 'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.88.jpg',
'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.89.jpg', 'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.9.jpg',
'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.90.jpg', 'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.91.jpg',
'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.92.jpg', 'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.93.jpg',
'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.94.jpg', 'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.95.jpg',
'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.96.jpg', 'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.97.jpg',
'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.98.jpg', 'F:/Face_Recognition_Attendence_System/TrainingImage\\Train.User.6.99.jpg']
```



```
.password_text.place(x=250, y=250)

.password_entry = Entry(self.login_frame_1, font=("times new roman", 15), bg="lightgrey", show="*")
.password_entry.place(x=250, y=280, width=350, height=35)

xt="Login", font=("times new roman", 20, 'bold'),
7", command=login)
height=40)


ock", font=("Book Antigua", 25, 'bold'), fg="white",
", bd=0)
=350)


'#800857', background='white',
```

```
6 Farhan
6 Farhan
6 Farhan
6 Farhan
6 Farhan
6 Farhan
6 Farhan
6 Farhan
6 Farhan
```

Face_Recognition_Attendence_System › main.py   211:18   CRLF   UTF-8   4 spaces   Python 3.11 (Face_Recognition_Attendence_System)   363 of 1506M

CONCLUSION

The conclusion is the final section of a report that summarizes the key findings, outcomes and recommendations of the project. In the context of a face recognition project using Python and OpenCV, the conclusion would summarize the results achieved, the challenges faced and the future scope for improvement.

In the conclusion, it is important to discuss the overall impact and relevance of the project in the field of computer vision and pattern recognition. This can include the comparison of the results obtained with the existing state-of-the-art solutions, the limitations of the current implementation and the potential for future developments.

Additionally, the conclusion should highlight the benefits of the project, such as improved accuracy, faster processing time, and enhanced user experience, to name a few.

In conclusion, the report should provide a concise and clear overview of the project's goals, results and impact, and provide valuable insights for future work in the field of face recognition.