Preface

It brings me great pleasure to deliver the project on the "**FaceTrek Attendance - Face recognition Attendance System**" that was prepared honestly, on time, and with the utmost care.

Technology has assimilated into our daily lives in the quickly changing world of today. In the world of identity verification and attendance tracking, technological improvements have had a significant impact. Traditional means of manually recording attendance, such as paper registers and swipe cards, take a lot of time and are prone to fraud and inaccuracies. The Face Recognition Attendance System, a cutting-edge system that uses facial recognition technology to simplify attendance management, is developed in response to these difficulties.

An important advancement in the way organizations, institutions, and corporations track and manage attendance is the Face Recognition Attendance System. Modern facial recognition algorithms and high-resolution cameras are used by this cutting-edge system to identify and confirm people's identities quickly and accurately. This technology provides a safe, contact less, and effective way to track attendance by utilizing the special and inborn characteristics of the human face.

The goal of this project is to give readers a thorough understanding of the Face Recognition Attendance System, from its fundamental ideas and parts to how it works. We explore the algorithms, hardware, and software that power facial recognition technology, delving into its deep nuances. We also go over the privacy issues and ethical issues surrounding the usage of facial recognition technology, highlighting the significance of careful deployment and data security.

Throughout this project, we will embark on a journey to design, create, and deploy a working Face Recognition Attendance System. We will discuss the numerous phases involved in constructing such a system, including data gathering, model training, system integration, and user interface design. Readers will have a thorough blueprint for building their own face recognition attendance system or receive insightful knowledge of the technology underlying this creative solution by the conclusion of this assignment.

The Face Recognition Attendance System is an example of how technological improvements have the potential to improve efficiency, accuracy, and security in attendance management. This project provides a road map for individuals looking to use facial recognition technology to completely transform how attendance is tracked and handled.

We believe that this initiative will encourage and enable people, groups, and institutions to investigate the potential of facial recognition technology and its revolutionary effects on attendance monitoring.

This project has undergone meticulous scrutiny to ensure that it is flawless and beneficial in every way.

Shubham Bhalerao

Acknowledgment

This Face Recognition Attendance System project's successful completion was made possible by the concerted efforts, leadership, and assistance of numerous people and organizations. I would like to sincerely thank everyone who contributed significantly to making this initiative a success.

First and foremost, I would like to express my sincere gratitude to Prof. Manoj Singh, my project supervisor, whose knowledge, support, and priceless insights significantly influenced the course of my work. Throughout the project, Prof. Manoj Singh gave me constant support and guided me through the challenges of facial recognition technology.

Additionally, I owe a debt of gratitude to the faculty, staff, and administration of SIES College of Arts, Science, and Commerce, whose expertise and resources proved to be crucial to my research and development endeavors. They have consistently served as an inspiration with their commitment to creating a supportive learning environment.

I would like to express my gratitude to the individuals who voluntarily agreed to take part in this study, since their participation was essential in gathering the information required for developing and validating my facial recognition model. The completion of this project would not have been feasible without their assistance.

The open-source community and the creators of the many libraries and technologies that I tapped into while creating my face recognition attendance system deserve recognition. I appreciate their dedication to advancing technology and their significant contributions.

Special thanks go out to my peers and friends for their emotional support and encouragement throughout the highs and lows of this endeavor. I was inspired to continue working by their zeal and faith in our job.

Finally, I want to thank my families from the bottom of my hearts for all their support, patience, and understanding. Their support and confidence in my talents have served as my inspiration.

I sincerely thank everyone who helped with this initiative, whether they were directly involved or not. Your assistance has been crucial in helping me realize my vision.

Mr. Shubham Bhalerao

SIES College of Arts, Science and Commerce

Date

Contents

Preliminary Investigation

Preliminary Investigation

Innovativeness

With no longer requiring manual tracking of attendance, the suggested method will reduce paperwork. The overall amount of time required to record attendance will be reduced by the new system.

To ensure the accuracy of the attendance data, the new system will collect individual attendance using facial recognition technology.

The prior system needs to be evolved to increase productivity, ensure data accuracy, and allow access to the information for those legitimate purposes.

Without human mistakes, facial recognition software can accurately manage time and attendance. With a facial recognition time monitoring system, you won't ever have to be concerned about time fraud or "buddy punching."

Employees may log in and out of the building faster with facial recognition because they don't have to contact the system's surface. This reduces the transmission of illnesses while also saving time.

The lack of authorization :- Facial recognition software only requires a clear image of the subject's face to work therefore data can be easily acquired in public locations.

Predatory marketing :- Some businesses may utilize software that analyzes facial expressions to prey on susceptible clients.

Identification fraud :- Criminals who have amassed sufficient personal data about you may engage in identity fraud.

In any setting, the system will mark and record attendance.

Since this system is automated, attendance will be recorded in accordance with that, greatly boosting accuracy, and a report on attendance will then be generated.

These are the key components of this technology:- 1) Face recognition 2) Face identification.

1) Face detection: A computer technology that recognizes human faces in digital photos, face detection is utilized in a range of applications.

2) Face recognition: A facial recognition system is a computer program that can recognize or authenticate a person from a digital image.

Steps to use FaceTrek Attendance on computer:

1. The representative or user of a specific department must first sign up for the system.
2. After that, he or she must enter valid credentials to access the system.
3. After proper validation, he or she will add the student, teacher, and staff data to the system.
4. After that, he or she will input the samples of their faces into the system.
5. After that, he or she will use the sample image data to train the facial recognition model.
6. The representative or user will then be prepared to take attendance using face recognition on each day of class or school.

Uses of FaceTrek Attendance:

- A proper confirmation message is given to every person (present or absent) by mail and standard SMS texts.
- Attendance is managed without human intervention.
- Reports or attendance records are automatically generated in CSV files.
- Create an invoice for each task that is successfully finished.
- Searching is permitted on all individual data (students, teachers, employees, and users from the admin side) via a straightforward, user-friendly GUI.
- Safeguarding attendance records with backups.

Benefits of FaceTrek Attendance:

- Less human mistake.
- Ease of maintenance.
- Time savings.
- Backup of attendance records.
- Invoice for successful operations.
- Confirmation messages to all users.
- Accelerate the work.
- No possibility of proxy.
- Attendance based on given or allotted period.

Existing System

- Students must manually enter data into the current system.
- Registers written by hand will be used to track attendance here.
- Keeping track of the user's record will be a laborious task.
- This requires more human effort.
- Since records are kept in manually written registers, retrieving information is difficult.
- This application needs accurate data entered the appropriate fields.
- If the incorrect inputs are made, the application might not function properly, making it challenging for the user to utilize.

Limitation of Existing System:

- It takes more time.
- Involves a lot of paperwork.
- Increases the possibility of human mistakes.
- Improperly manages attendance records.
- Is less accurate.
- Increases the likelihood of proxy.
- Lack of a certification of attendance for the person.
- Data collision risks across classes.

Risk Management

Lack of consent:

- Facial recognition software only needs a clear picture of the subject's face to start collecting data, this can be done simply in public settings.

Predatory marketing:

- Software that analyzes facial expressions may be used by some businesses to engage in predatory marketing, preying on weak clients.

Identity fraud:

- Identity fraud is a crime that can be committed by criminals who have gathered enough personal information about you.

Security Issues:

- Beware of attacks on the facial recognition attendance system using print photos, smartphone photos, replays, photo masks and 3D masks attack.

Objectives

- A software program called an attendance management system was created to ensure that students, teachers, and staff show up to class each day.
- The staff members in charge of the department oversee recording the individuals' attendance in this instance.
- After recording each person's attendance and any absences, notification verification is carried out.
- When an unidentified person is recognized and for all successfully executed operations, an invoice is created.
- Attendance is done at predetermined intervals, such as for a person for only six or twelve months, and then when that person appears in front of the camera, my model will depict them as an unknown person.

Proposed System

The proposed system is evolved to address the shortcomings of the current system. The proposed system in a Face Recognition Attendance System project normally comprises of several different parts and technologies that cooperate to produce the intended functionality. The following are the essential elements of the FaceTrek Attendance-Face Recognition Attendance System:

Data Collection and Preprocessing:

- Image preprocessing techniques to improve image quality, such as noise reduction and face alignment; high-resolution webcams for obtaining facial images.

Face Detection:

- A face identification method to find and extract faces from the collected photos (Haar cascades).

Face Recognition Model:

- Using a varied dataset of facial photos of well-known people to train the model.

Database Management:

- A database to keep track of registered users' attendance history and facial template data. I have utilized the MongoDB NoSQL database.

User Registration Module:

- A module for adding new users, taking their photos, and putting their biometric templates in a folder.

User Interface:

- An intuitive user interface that allows users and administrators to communicate with the system.

The Present system works by obtaining 100 image samples of an individual's face and then identifying it based on those 100 image samples. It then saves the attendance records in a CSV file and sends an email and SMS to the person as confirm their attendance.

The user must first register in the system, receive an OTP and a confirmation email via email, and then log in to store student, teacher, and staff data as well as to identify them and take attendance.

The system will be fully under the admins' control. In addition, he will save the CSV attendance file to another folder and automatically produce a new CSV file for the following day's attendance. He may change the details of every individual, mark attendance, and send emails and SMS to those who are absent. Additionally, each individual window has a search system for conducting searches.

Here, my project attempts to cut down on paperwork and save time to produce accurate attendance results. The system offers the best user interface, and improper behavior such as proxy is not tolerated. The suggested system can be used to produce effective reports. With minimal time spent, we can obtain data whenever it is needed.

Advantages of Proposed System

- Simple to manage.
- User-friendly
- Sending SMS and emails of confirmation
- Saving attendance information as a CSV fie
- Speed
- Invoice for each activity that is successfully completed.
- Appropriate input field validation.
- Attendance depending on the allowed or designated time.

Tools and Model

- Python-based libraries such as tkinter, re, time, datetime, pymongo, os, shutil, csv, gtts, playsound, twilio.rest,tkcalendar,PIL,cv2,numpy.
- Agile Model. It is a flexible and iterative approach that allows for ongoing adaption to change requirements and priorities.

Modules (Functional Requirements):

- Admin Module.
- User Registration and Enrollment Module.
- Face Recognition Module.
- Face Detection Module.
- Attendance Tracking Module.
- User Interface Module.
- Camera Interface Module.
- Notification Module.
- Database Module.
- Data Storage Module.
- Machine Learning Model Training Module.

Non – Functional Requirements:

Non-functional requirements are limitations that must be followed when developing an application.

Usability:

- The clients or stakeholders will use the program or product that I am producing.

Efficiency:

- Time complexity is decreased by my software's ability to complete tasks more quickly, such as issuing notifications and recording attendance.

Reliability:

- The program I'm working on is intended to deliver a range of services as anticipated by the stakeholders.

Maintainability:

- The software I'm creating will have high performance features, such as manual data updates without losing previously stored data.

Maintenance and Estimated Costs

Costing and other fees are added in accordance with the company's regulations. Pricing can be adjusted to meet new needs.

- Price (per month salary) : Rs. 60,000
- 6 months' salary : Rs. 3,60,000

LIST	COST
Hardware System Requirement	1,90,000
Software System Requirement	2,70,000
Designing Charge	4,50,000
Functionality Charge	5,60,000
Designer & Developer Team Salary	3,60,000 (for six months)
Company Profit	1,40,000
Maintenance Charges	50,000 (per month)
Miscellaneous Charges	70,000
Tax (10%)	93,000
Total Amount	**2,083,000**

System Requirements

Hardware Requirements (Minimum) :

Processor:- Intel i3 Processor and above.

Primary Memory:- 4GB RAM (Random Access Memory) and above.

Storage:- 8GB hard disk and above.

Software Requirements:

Operating System:- Microsoft Windows (7,8, and 10), Linux, and Mac OS.

Software:- Python IDLE.

Software Requirement for Development of FaceTrek Attendance:

Front-end:- Python (tkinter)

Package:- tkinter, re, time, datetime, pymongo, os, shutil, csv, gtts, playsound, twilio.rest,tkcalendar,PIL,cv2,numpy.

Back-end:- Python (pymongo).

Database:- MongoDB.

Software:- Python IDLE, MongoDB Compass.

Feasibility Study

Once the issue is properly identified, a feasibility study is carried out. The goal is to decide on a solution to an issue fast and cheaply. Details of operation and management are included.

The system has undergone tests to determine its viability in the following areas:

Project Scope and Objectives:

- The major goal of this automated attendance system for face detection and recognition is to offer face recognition in a real-time environment so that employees, teachers, and students can observe and mark their daily attendance so that their presence can be kept track of.

Market Analysis:

- The schools/colleges that are having trouble managing attendance on paperwork are the system's target market.
- I investigated the current attendance face recognition systems and tried to outperform them.

Technical Feasibility:

- This technique will be highly beneficial for recording attendance using facial data, reducing the likelihood of proxy scenarios.

Technical Requirements:

- The only technical requirements are a competent laptop or computer with fair internet connectivity and a webcam.

Resource Availability:

- The individual or user who will utilize the system should have a basic understanding of computers and operating systems.
- If the computer or laptop lacks speakers or a camera, additional external speakers and webcams can be required.

Stakeholders

- Admin
- Users
- Software Developer
- Students (Using via Face Detection)
- Teachers (Using via Face Detection)
- Staffs (Using via Face Detection)
- Unknown Persons (Using via Face Detection)

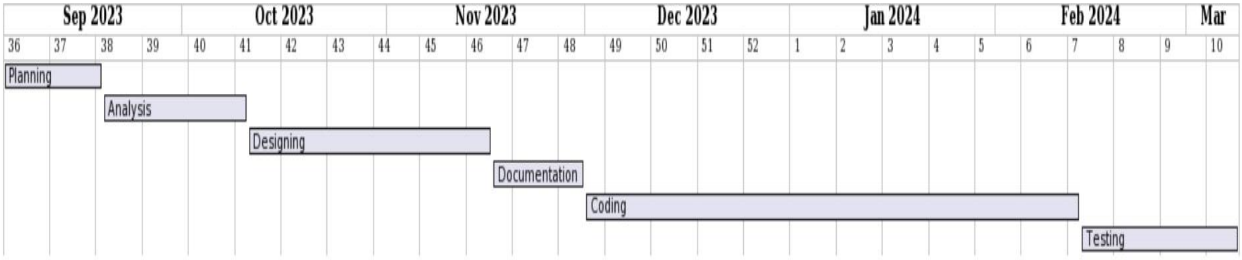
Events

1. Admin
    - Login
    - Logout
    - Add user information.
    - Update users' data
    - Eliminate users' data.
    - Refresh admin dashboard
    - Check out Students data.
    - Access Staffs data
    - Access Teachers data
    - Full system access
    - Run a user data search.
    - Set all input fields to their default values.
    - Notifying absent people via SMS and email with a confirmation message.
    - System reset for attendance the following day.

2. Users
    - Self-register
    - Verify with OTP
    - User Login
    - User Logout
    - Add individuals data.
    - Update individuals data
    - Delete individuals data.
    - Refresh each panel or window separately.
    - Conduct individual data searches.
    - Restore all input fields to their initial settings.
    - Take/update individual photo samples.
    - Develop the facial recognition algorithm.
    - Record attendance of people
    - Export the CSV file for attendance.
    - Import the CSV file for attendance.
    - Update the CSV file for attendance.
    - Looking at samples of individuals photos
3. Software Developer
    - Monitoring and support
    - Updates and maintenance
    - Data Management
    - User feedback
4. Students
    - Appear in front of the camera and record your attendance.
5. Teachers
    - Appear in front of the camera and record your attendance.
6. Staffs
    - Appear in front of the camera and record your attendance.
7. Unknown persons
    - Appear in front of the camera.

Gantt Chart

Planned



Phase Title	Start Date	End Date	No. Of Days	Signature
Planning	04/09/2023	18/09/2023	15	
Analysis	19/09/2023	10/10/2023	22	
Designing	11/10/2023	16/11/2023	37	
Documentation	17/11/2023	30/11/2023	14	
Coding	01/12/2023	13/02/2024	75	
Testing	14/02/2024	08/03/2024	24	

System Analysis

Fact-Finding Technique

Interview - Based :- The interviews I conducted with the Head of Department (HOD), Principal, Teachers, Staff, Students, and other stakeholders gave me a great understanding of how important it is for them to have an attendance system that uses facial recognition and how the various attendance procedures work. The information I acquired greatly aided them in meeting their needs and simplifying their work.

Documentation and Record View :- This kind of review assists us in learning about the procedures used while taking attendance, such as how the college as a whole manages attendance, how courses are handled when faculty members are absent, and what challenges they encounter.

Event Table

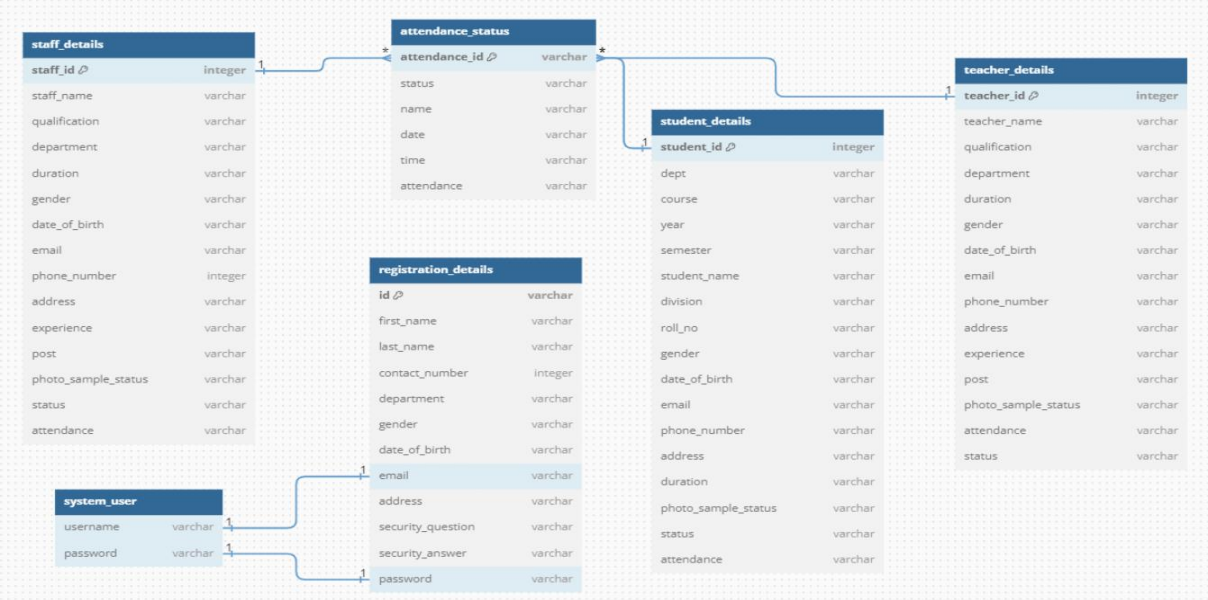
Sr no.	Event	Trigger	Source	Activity	Response	Destination
1.	Admin Login	Admin logs in	Admin	Input username and passwords	Admin dashboard or Error	Admin
2.	Admin Logout	Admin logs out	Admin	Click logout button	Login window	Admin
3.	Add user information	Add user information details	Admin	Enter User Details	User Added or Error	Admin, User
4.	Update user's data	Update user information details	Admin	Enter User Updated Details	Details Updated or Error	Admin, User
5.	Eliminate user's data	Delete user information details	Admin	Enter users date of birth	User deleted or Error	Admin, User
6.	Refresh admin dashboard	Admin dashboard gets refreshed	Admin	Click the refresh button	Dashboard refreshed	Admin
7.	Check out students' data	Viewing student's full database details	Admin	Click the student data button	Student's data window	Admin
8.	Access teachers' data	Viewing teacher's full database details	Admin	Click the teacher's data button	Teacher's data window	Admin

9.	Access staffs' data	Viewing staff's full database details	Admin	Click the staff data button	Staff's data window	Admin
10.	Full system access	Access to the FaceTrek Attendance system	Admin	Click the home button	FaceTrek Attendance System	Admin
11.	Run a user data search	Perform search on user data with parameters	Admin	Choose parameter from combo box input detail and click search button	Searched user data or No records found	Admin
12.	Set all input fields to their default values	Setting fields to their default values	Admin	Click the reset button	Fields reset to their default values	Admin
13.	Notifying absent people via SMS and email with a confirmation message.	Sending absent notifications	Admin	Click the Absent Message button	Notifications sends successfully or Error	Admin, teachers, students, staffs
14.	System reset for attendance the following day	Saves CSV to another folder, new CSV file is created, and the attendance fields of all individuals is set to absent.	Admin	Click Reset Data button	CSV file saved, a new CSV file is created, and all attendance fields of individuals is set to absent in database.	Admin
15.	Self – registered	Add details to register	User	Enter proper details, use strong password.	User is registered Successfully or Error occurred.	User
16.	Verify using OTP	Verify OTP to register	User	Verify OTP for completing registration process.	User is registered Successfully or Error occurred.	User
17.	User Login	User Logs in	User	Enter username and password	Successful Login or Error	User
18.	User Logout	User Logs out	User	Clicks exit button	Successful Logouts	User
19.	Add individuals data	Add staff, student, and teacher	User	Enter staff, student, and teacher details	Details added or Error	User

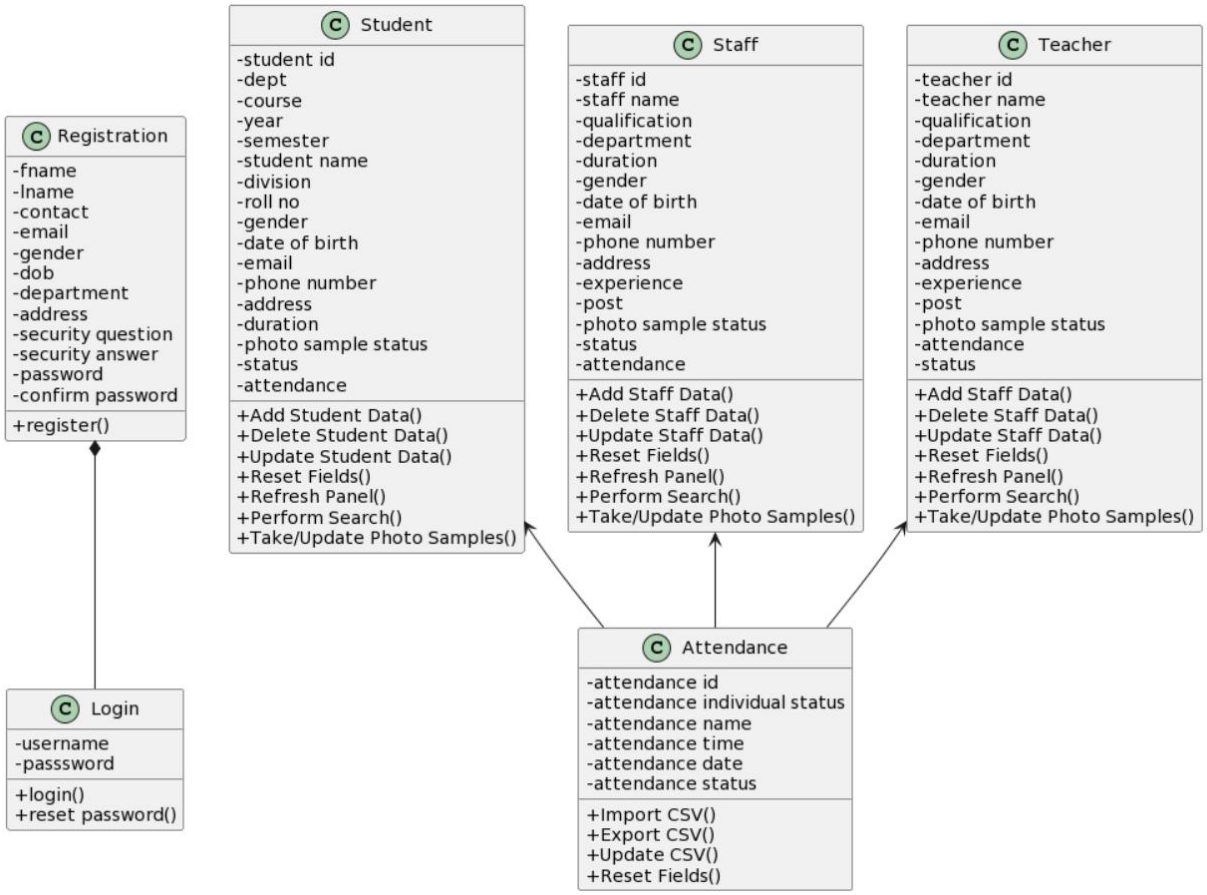
		information				
20.	Update individuals data	Update staff, student, and teacher information	User	Enter staff, student, and teacher details	Details updated or Error	User
21.	Delete individuals data	Delete staff, student, and teacher information	User	Id of individuals must be entered	Details deleted or Error	User
22.	Refresh each panel or window separately.	All individual windows/panel gets refreshed	User	Clicks refresh button	Individuals panel/window gets refreshed	User
23.	Conduct individual data searches.	Perform search on individual data with parameters	User	Choose parameter from combo box input detail and click search button	Searched user data or No records found	User
24.	Restore all input fields to their initial settings.	Setting fields to their default values	User	Click the reset button	Fields reset to their default values	User
25.	Take/update individual photo samples.	Take/Update photo samples	User	Click the take/update photo sample button	Photo samples taken/updated	User
26.	Develop the facial recognition algorithm.	Train the face recognition model	User	Click Train Data button	Model is trained with 100 photo samples of everyone	User
27.	Record attendance of individuals	Take Attendance	User	Click Recognize button	Attendance is taken or Error	User, students, teachers, staffs
28.	Export the CSV file for attendance	Export the updated CSV attendance file	User	Click Export button	File exported successfully or Error	User
29.	Import the CSV file for attendance	Import the attendance CSV file	User	Click Import button	File imported successfully or Error	User
30.	Update the CSV file for attendance	Update the imported CSV file	User	Update attendance details and click update button	Attendance CSV file updated successfully or Error	User

31.	Looking at samples of people's photos	Browse through 100 photo samples of individuals	User	Click Photos button	Sample Photos folder opens	User
32.	Monitoring and support	Monitors the system	Software Developer	Checks the system	System checking	Software Developer
33.	Updates and Maintenance	Provide updates to system	Software Developer	Check for updates	Updates done	Software Developer
34.	Data Management	Manage the data of stakeholders and database	Software Developer	Routine check	Data management	Software Developer
35.	User feedback	Take feedback from user	Software Developer	Take feedback via system or forms	Feedback taken	Software Developer
36.	Appear in front of the camera and record your attendance.	Give Attendance	Students	Comes in front of camera	Attendance done or Error	Students
37.	Appear in front of the camera and record your attendance.	Give Attendance	Teachers	Comes in front of camera	Attendance done or Error	Teachers
38.	Appear in front of the camera and record your attendance.	Give Attendance	Staffs	Comes in front of camera	Attendance done or Error	Staffs
39.	Appear in front of the camera.	----------	Unknown Persons	Comes in front of camera	Unknown face detected	Unknown Persons

ER Diagram

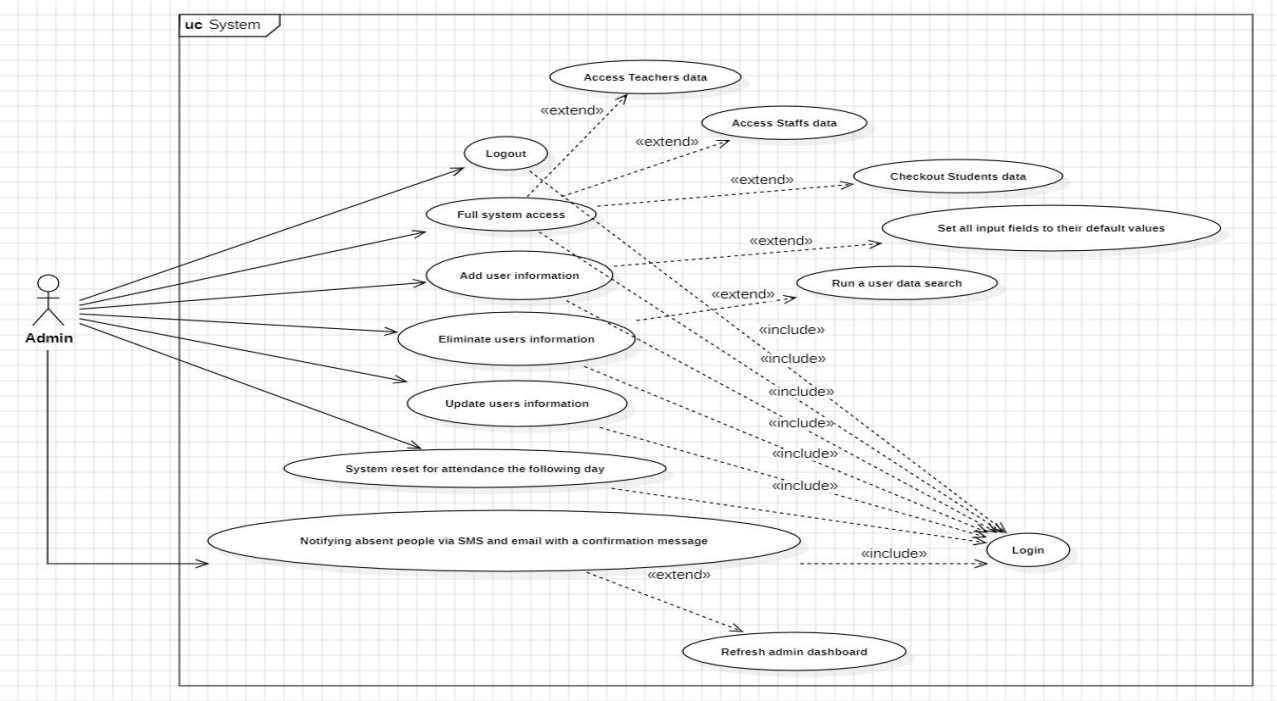


Class Diagram

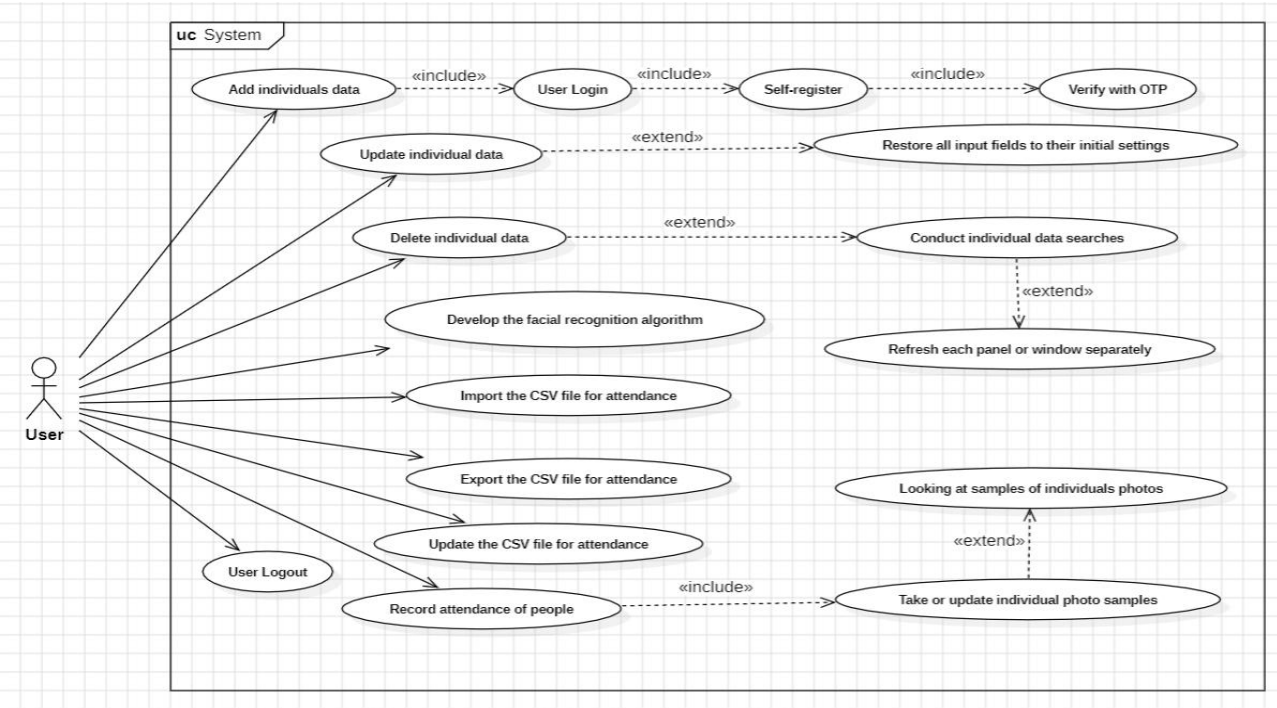


Use Case Diagrams

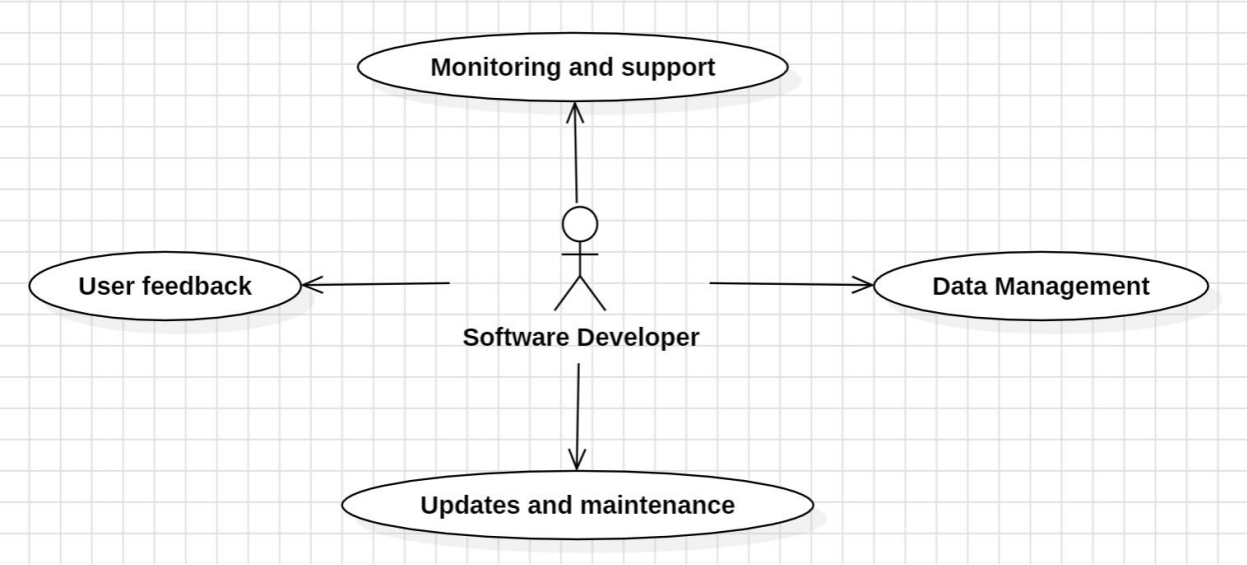
1) Admin



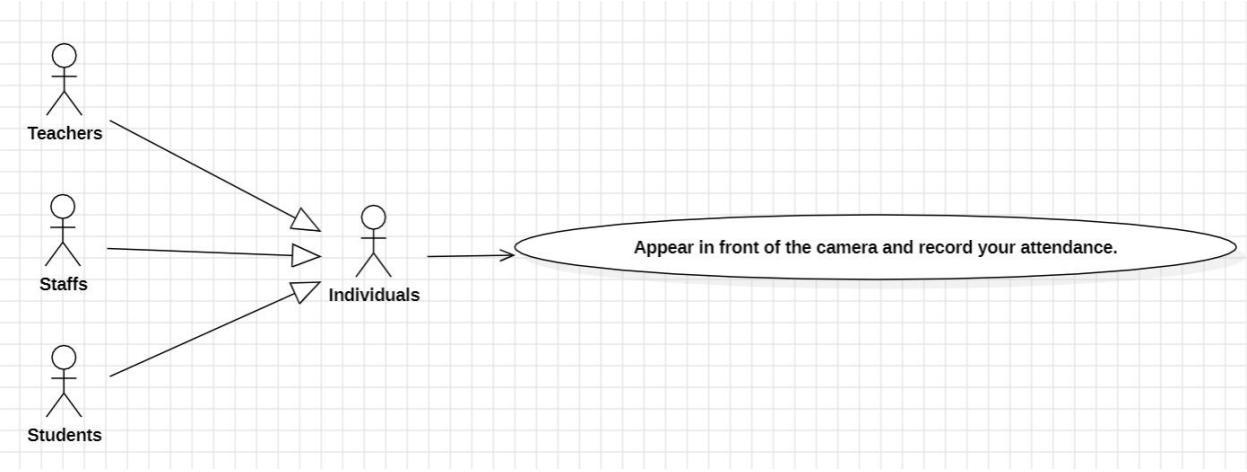
2) User



3) Software Developer

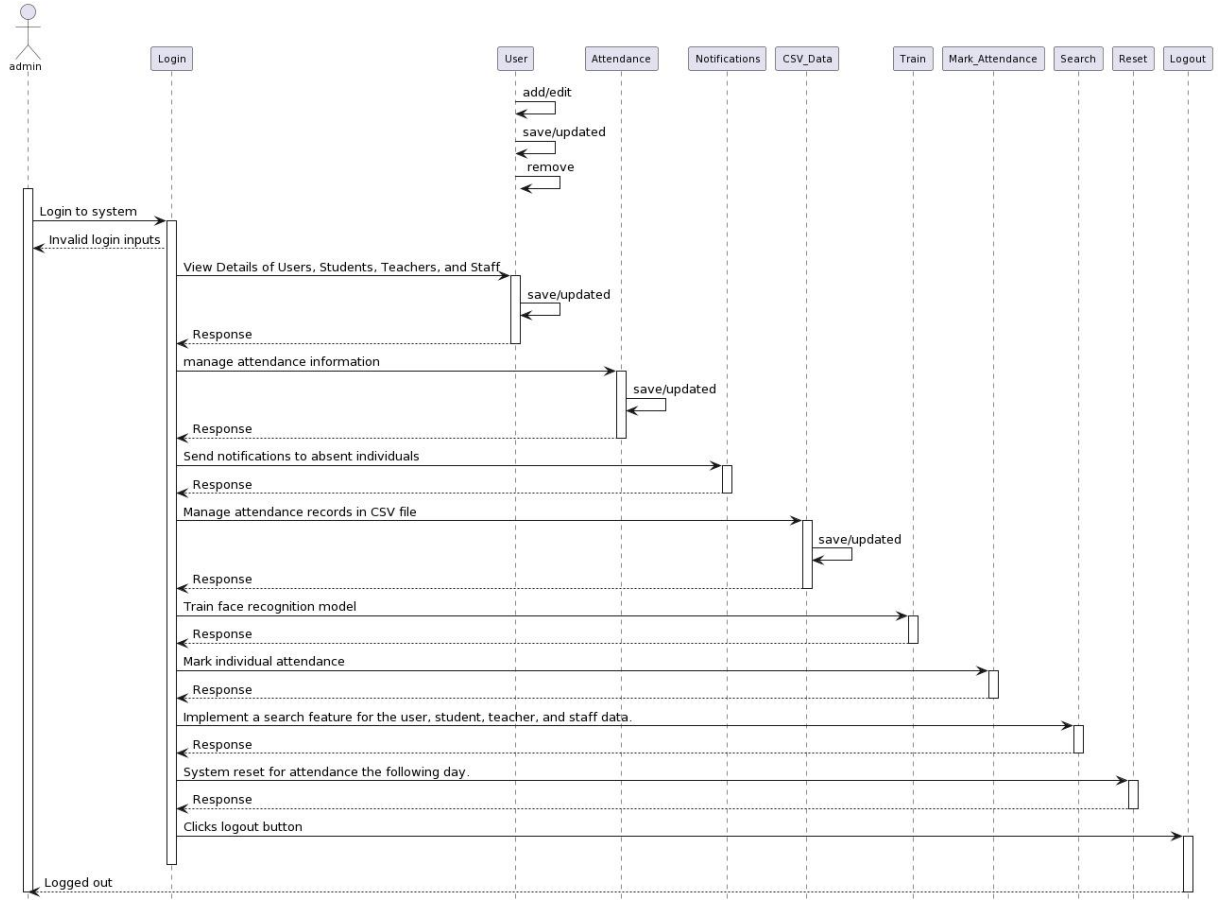


4) Individuals (Staffs, Teachers, Students)

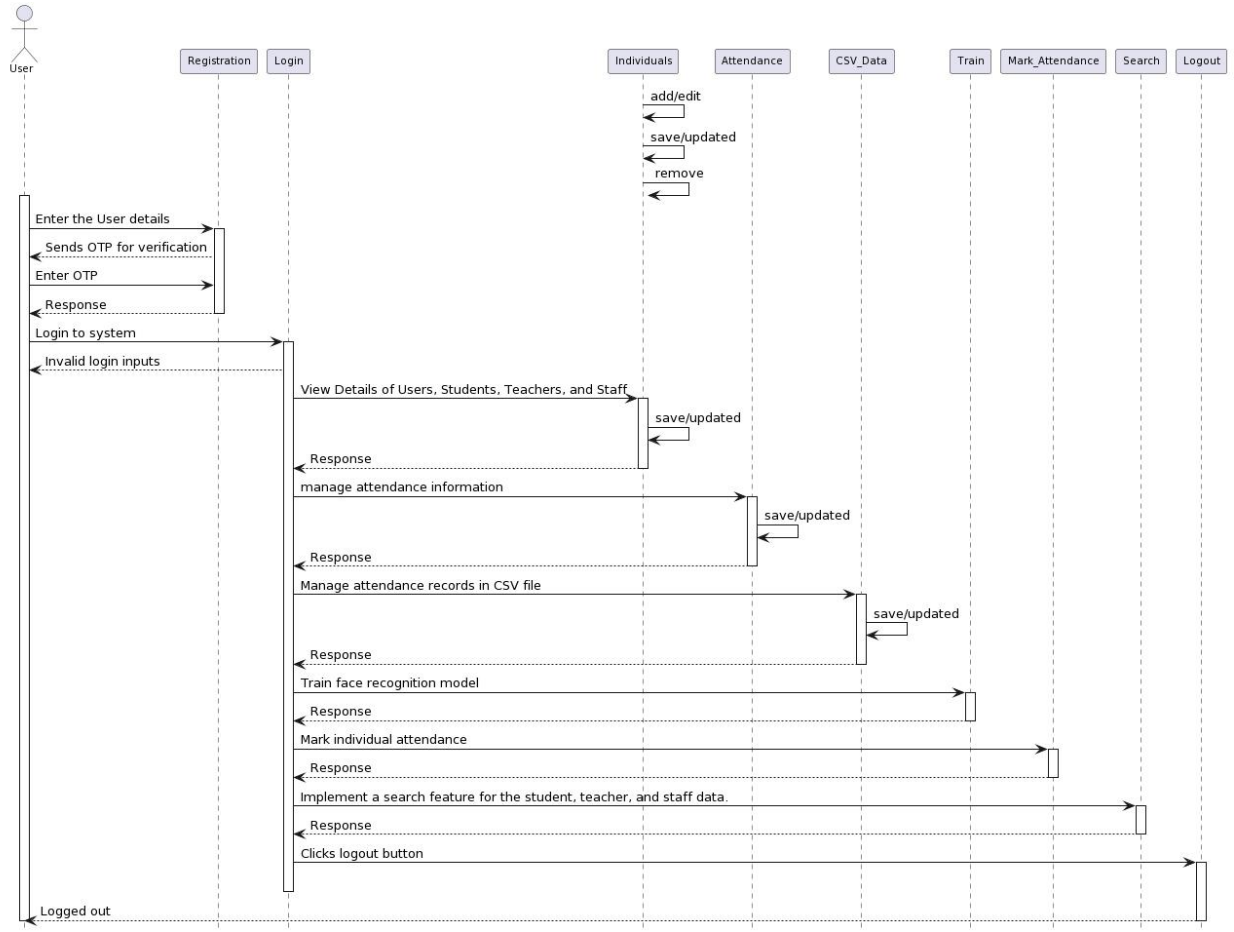


Sequence Diagrams

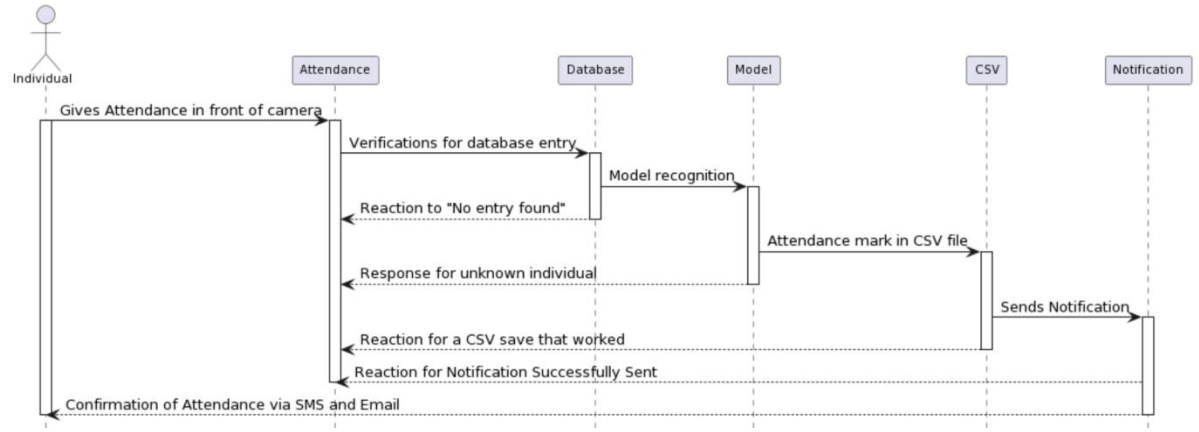
1) Admin



2) User

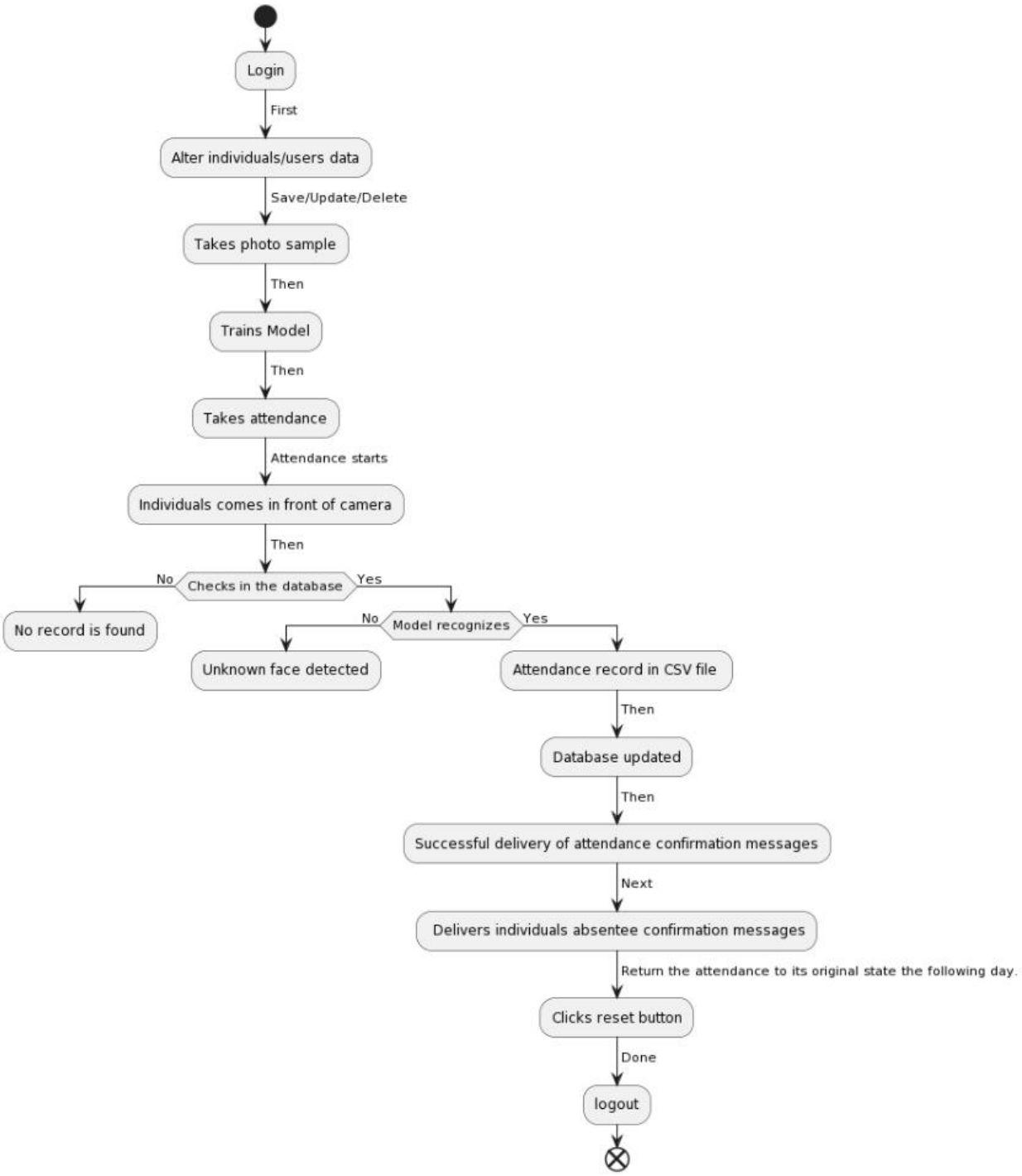


3) Individual

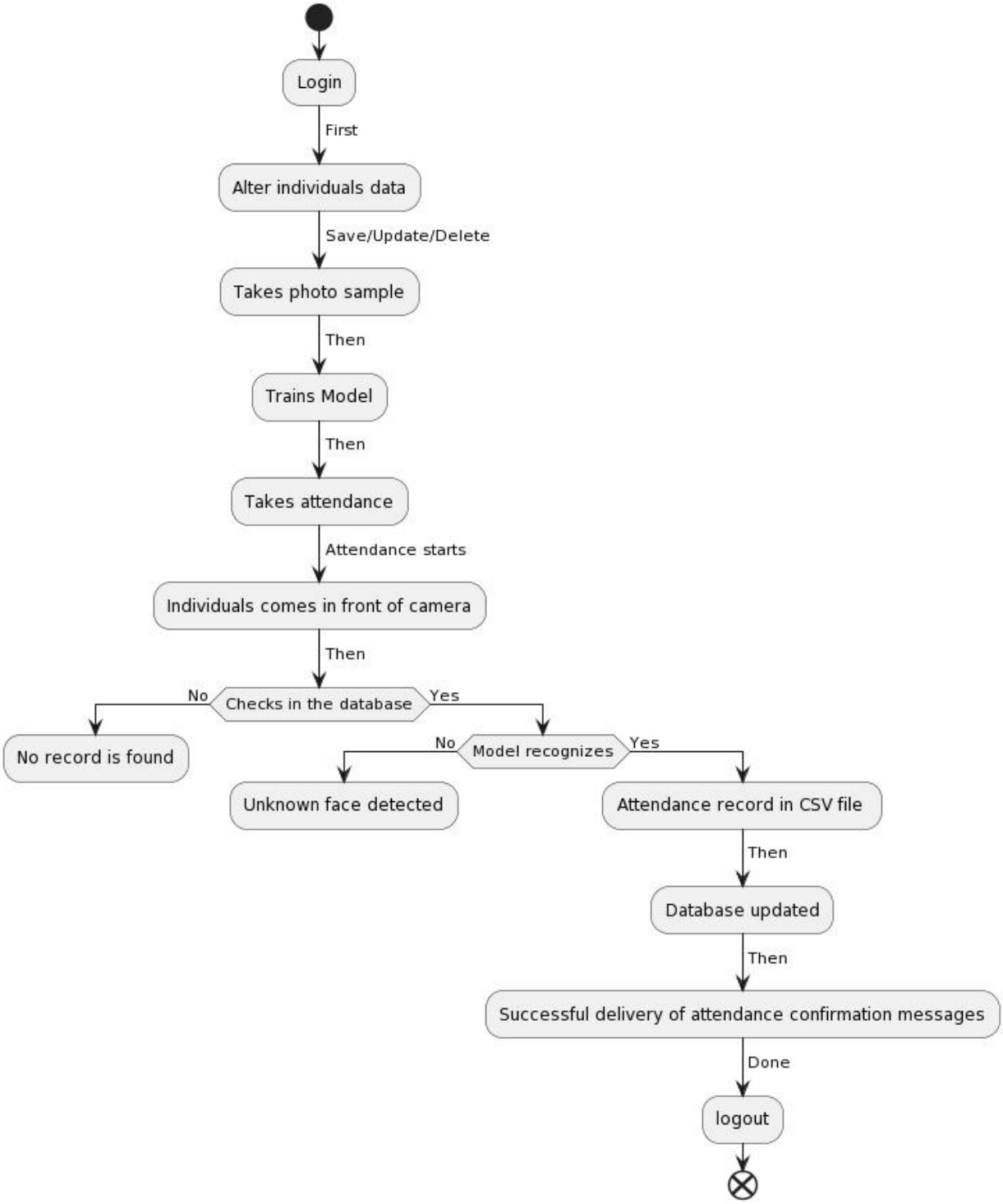


Activity Diagram

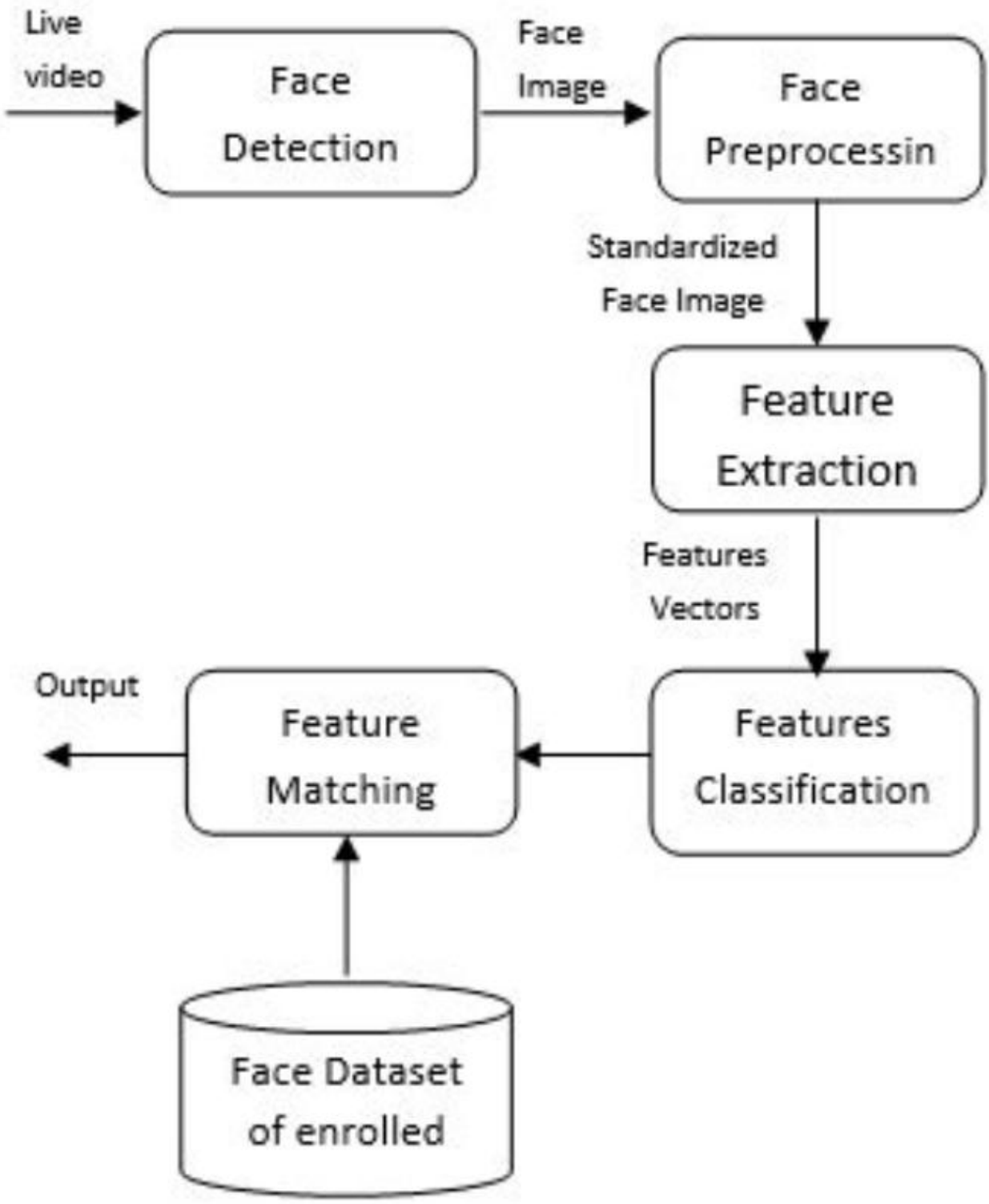
1) Admin



2) User


Login
First
Alter individuals data
Save/Update/Delete
Takes photo sample
Then
Trains Model
Then
Takes attendance
Attendance starts
Individuals comes in front of camera
Then
Checks in the database
No
Yes
No record is found
Model recognizes
No
Yes
Unknown face detected
Attendance record in CSV file
Then
Database updated
Then
Successful delivery of attendance confirmation messages
Done
logout

System Design



Conclusion

In conclusion, the implementation of a "**FaceTrek Attendance - Face recognition Attendance System**" represents a significant leap forward in the realm of workforce management. This technology not only streamlines the traditional attendance tracking process but also enhances security and accuracy. By leveraging facial recognition, organizations can mitigate the challenges associated with manual attendance systems, such as time fraud and buddy punching.

The face-based attendance system offers a user-friendly and non-intrusive method for employees to clock in and out, promoting efficiency and saving valuable time. Its integration with advanced algorithms ensures a high level of accuracy, reducing the likelihood of errors that may occur with traditional methods.

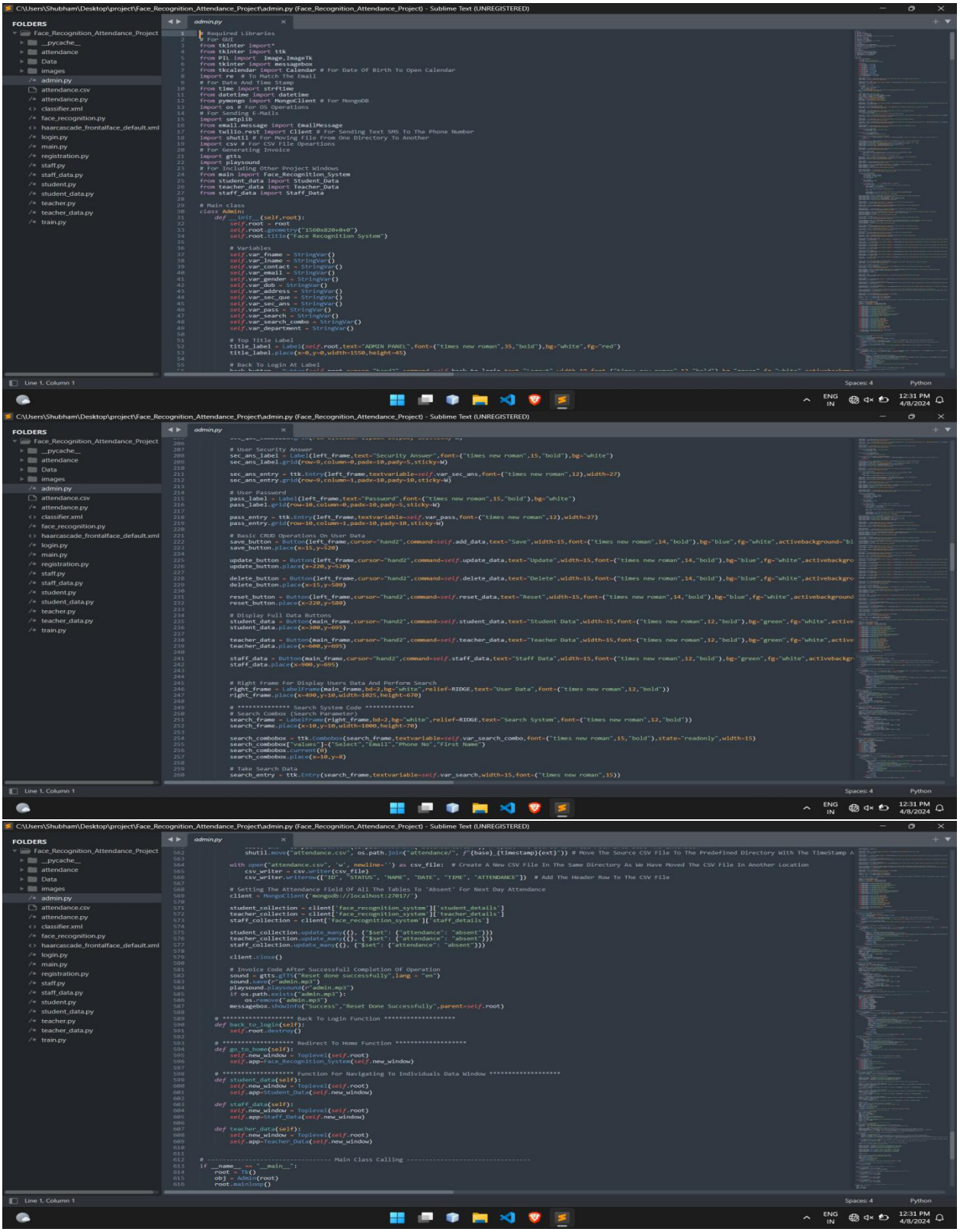
Moreover, the system contributes to a safer and healthier workplace environment by minimizing physical contact with attendance devices, a particularly important consideration in times of health concerns or global crises. The automation provided by "**FaceTrek Attendance - Face recognition Attendance System**" allows HR departments to focus on more strategic tasks, fostering productivity and strategic planning within the organization.

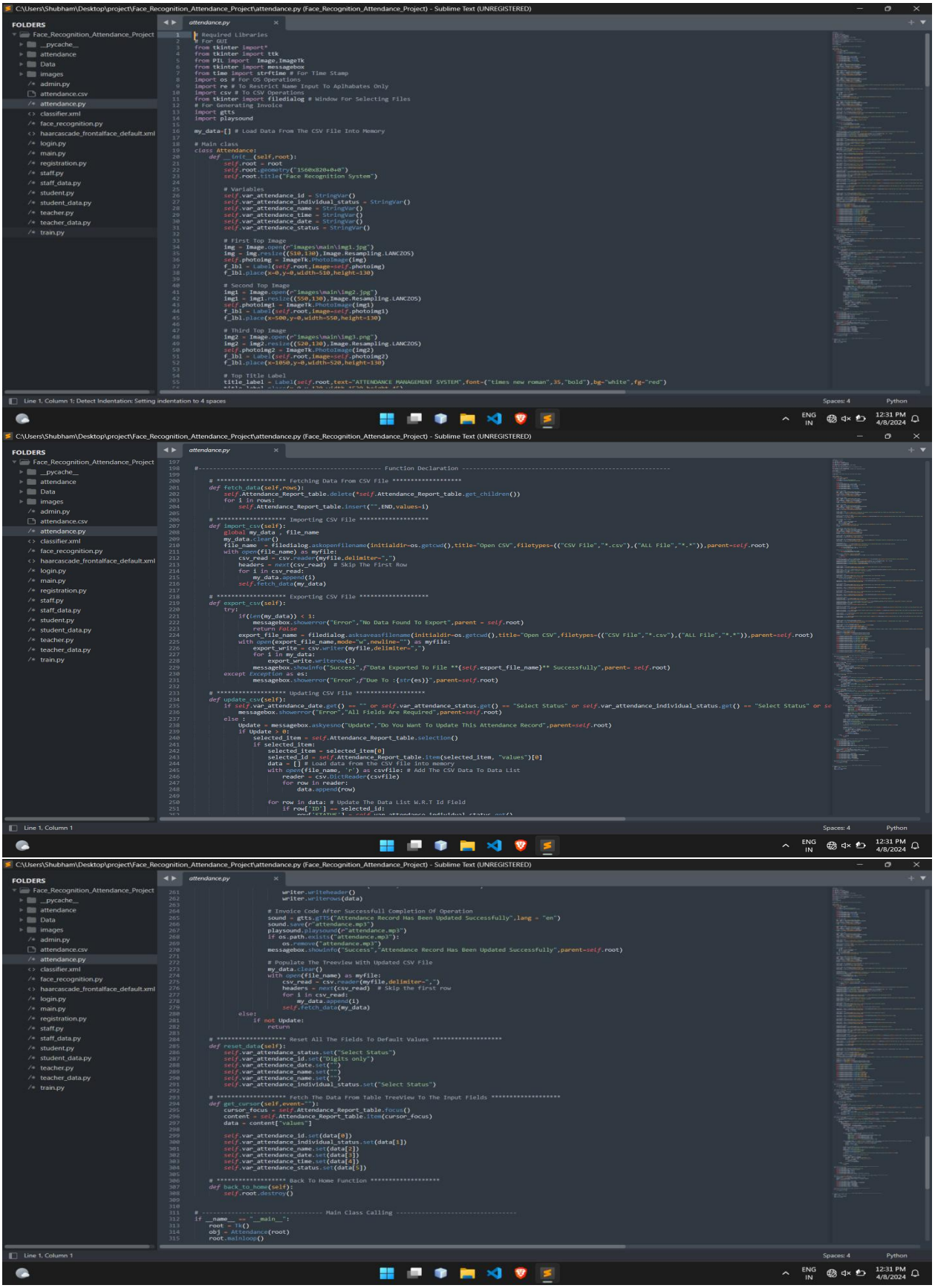
As we move towards an increasingly digitized era, "**FaceTrek Attendance - Face recognition Attendance System**" not only reflect technological advancement but also underscore a commitment to modern and efficient workforce management practices. Embracing this technology can lead to improved organizational processes, increased transparency, and ultimately, a more engaged and satisfied workforce.

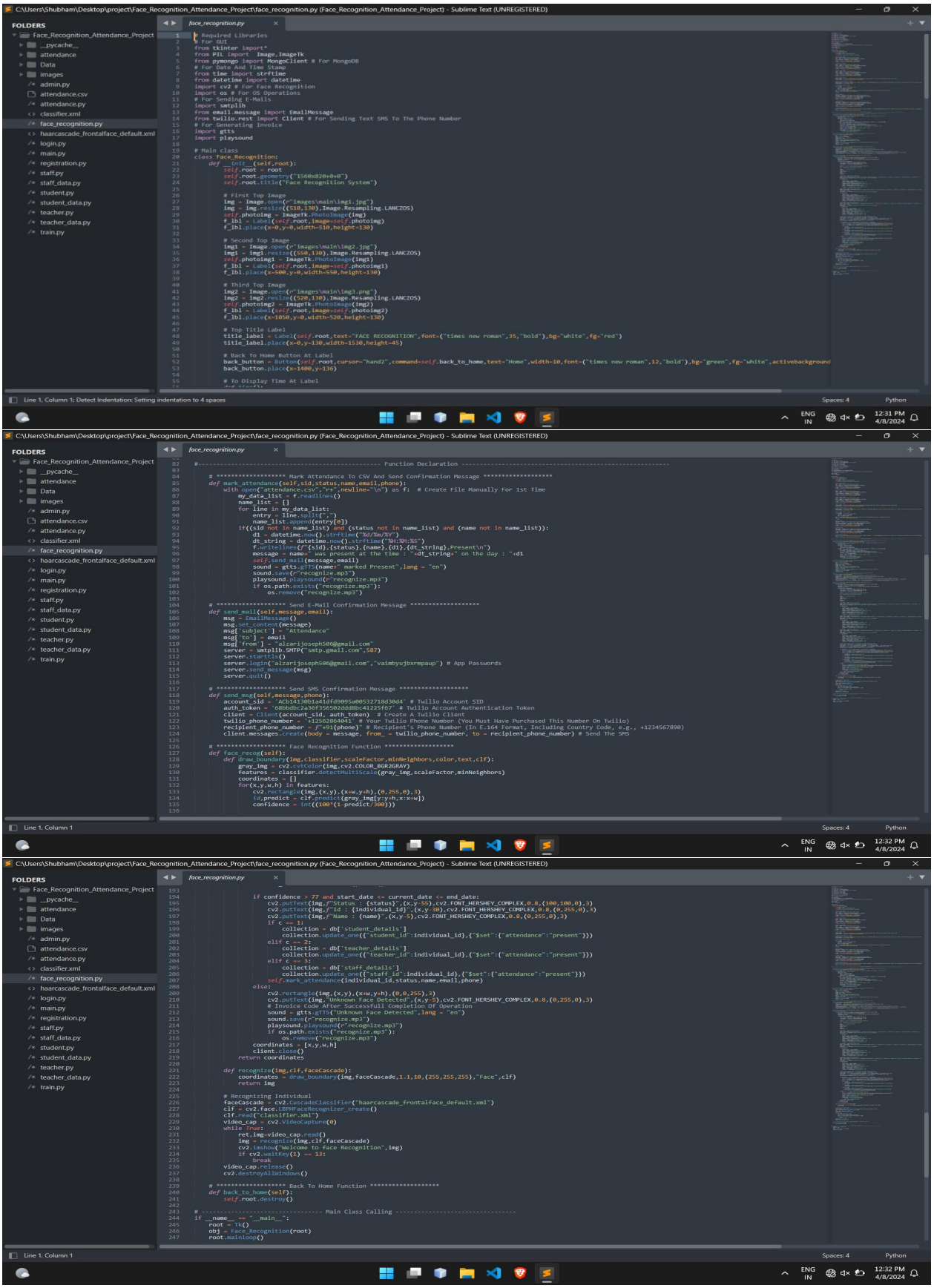
During the documentation phase, I made plans for the "**FaceTrek Attendance - Face recognition Attendance System**" design. Additionally, I researched the recognition model I would utilize for this project and came to the conclusion that the Haar Cascade Face Recognition Algorithm would yield the greatest results overall. Additionally, I've learned about a variety of Python packages that I might use for my project and database needs. I've made an effort to enhance the system's functionality to make it more dependable and interactive.

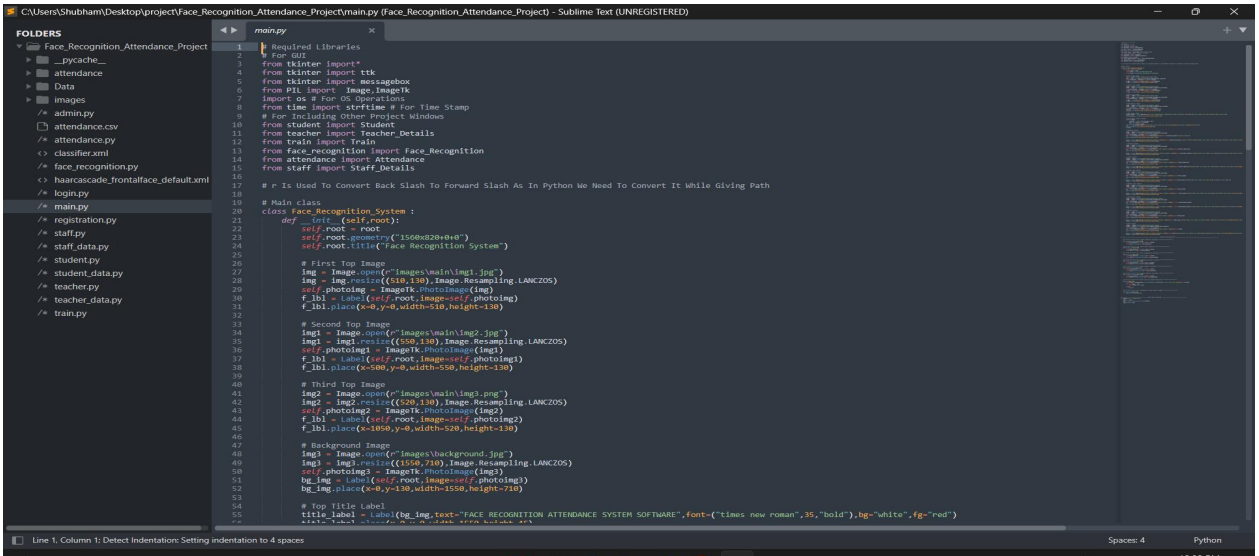
This system may encounter numerous difficulties, but I'm willing to adapt it further in order to get expertise from the industry and strive to make this project even better.

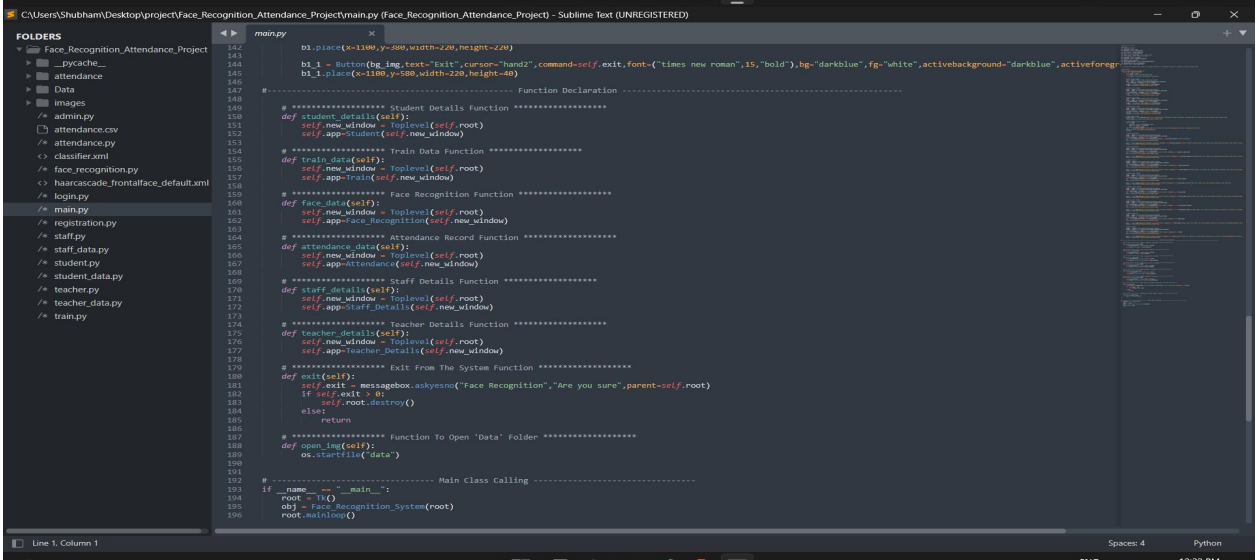
System Implementation (Code)

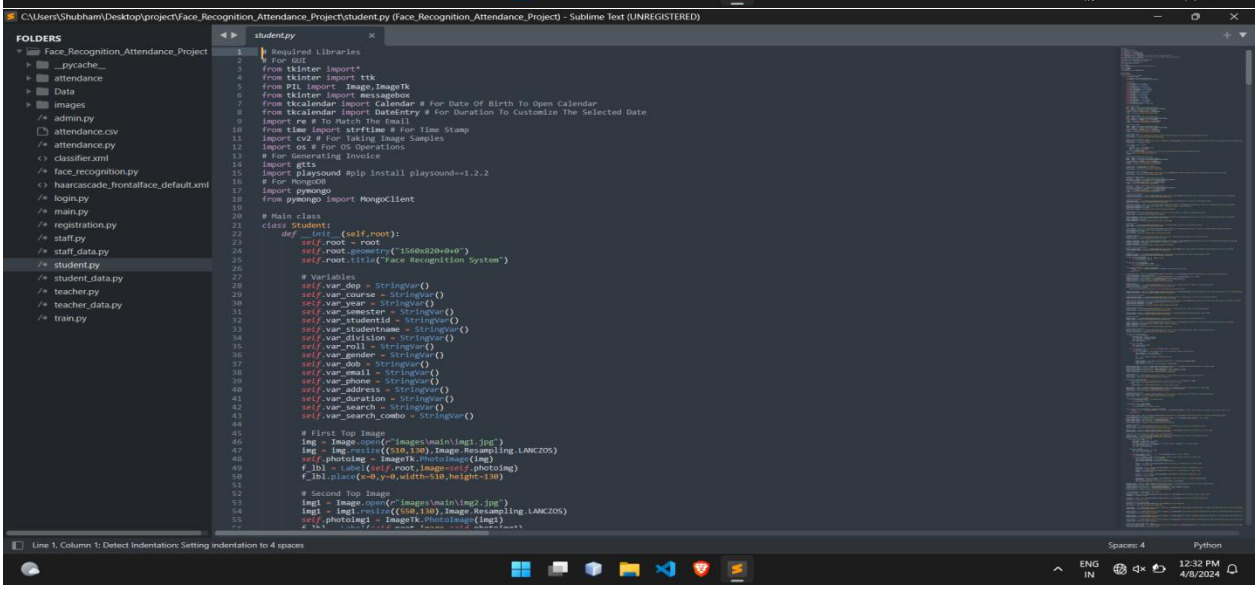


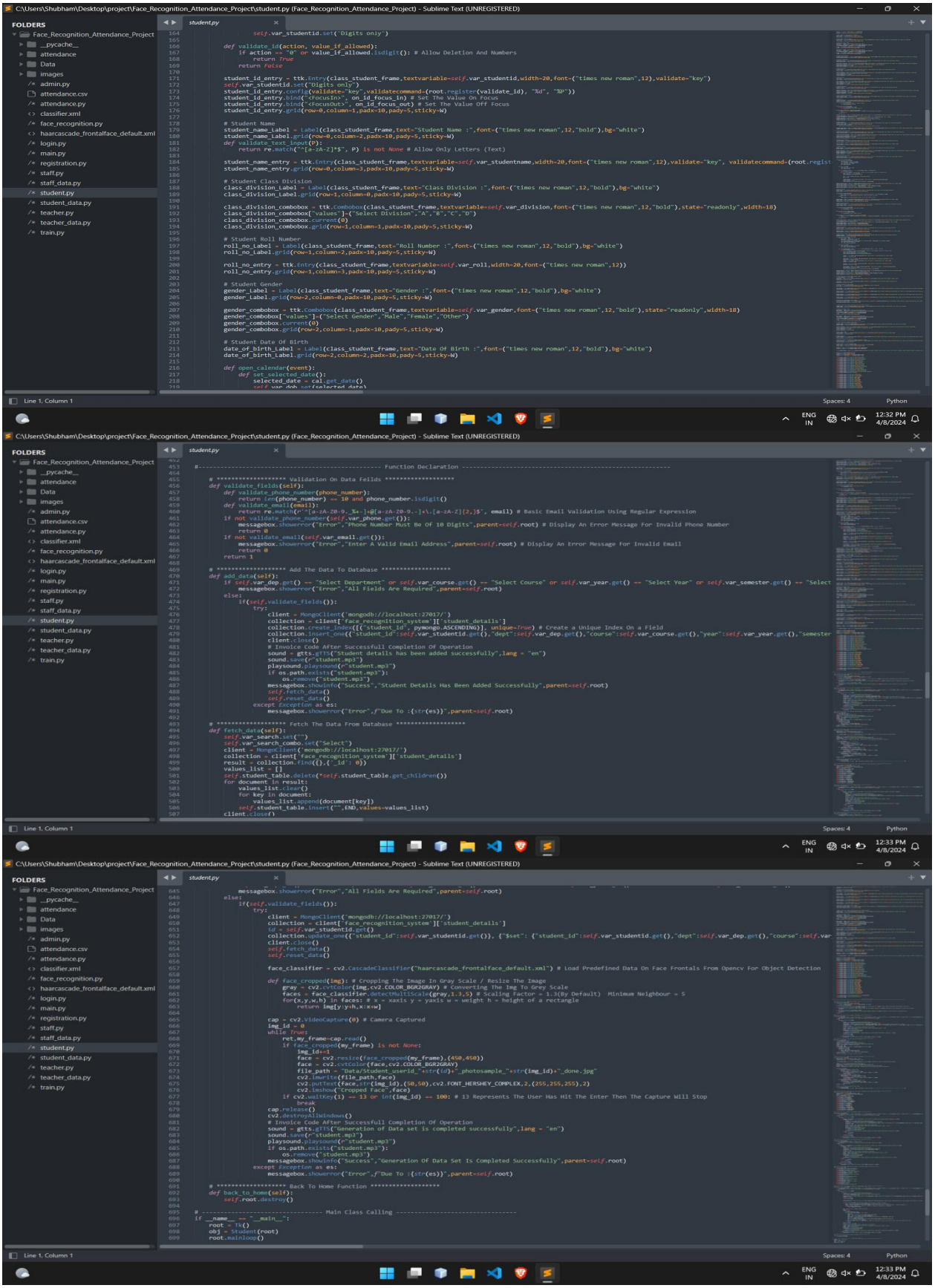

C:\Users\Shubham\Desktop\project\Face_Recognition_Attendance_Project\attendance.py (Face_Recognition_Attendance_Project) - Sublime Text (UNREGISTERED)
FOLDERS
Face_Recognition_Attendance_Project
__pycache__
attendance
Data
images
admin.py
attendance.csv
attendance.py
classifier.xml
face_recognition.py
haarcascade_frontalface_default.xml
login.py
main.py
registration.py
staff.py
staff_data.py
student.py
student_data.py
teacher.py
teacher_data.py
train.py
attendance.py
Required Libraries
For GUI
from tkinter import*
from tkinter import ttk
from PIL import Image,ImageTk
from tkinter import messagebox
from time import strftime # For Time Stamp
import os # For OS Operations
import re # To Restrict Name Input To Aplhabates Only
import csv # To CSV Operations
from tkinter import filedialog # Window For Selecting Files
# For Generating Invoice
import gtts
import playsound
my_data=[] # Load Data From The CSV File Into Memory
# Main Class
class Attendance:
def __init__(self,root):
self.root = root
self.root.geometry("1560x820+0+0")
self.root.title("Face Recognition System")
# Variables
self.var_attendance_id = StringVar()
self.var_attendance_individual_status = StringVar()
self.var_attendance_name = StringVar()
self.var_attendance_time = StringVar()
self.var_attendance_date = StringVar()
self.var_attendance_status = StringVar()
# First Top Image
img = Image.open(r"images\main\img1.jpg")
img = img.resize((510,130),Image.Resampling.LANCZOS)
self.photoimg = ImageTk.PhotoImage(img)
f_lbl = Label(self.root,image=self.photoimg)
f_lbl.place(x=0,y=0,width=510,height=130)
# Second Top Image
img1 = Image.open(r"images\main\img2.jpg")
img1 = img1.resize((550,130),Image.Resampling.LANCZOS)
self.photoimg1 = ImageTk.PhotoImage(img1)
f_lbl = Label(self.root,image=self.photoimg1)
f_lbl.place(x=500,y=0,width=550,height=130)
# Third Top Image
img2 = Image.open(r"images\main\img3.png")
img2 = img2.resize((520,130),Image.Resampling.LANCZOS)
self.photoimg2 = ImageTk.PhotoImage(img2)
f_lbl = Label(self.root,image=self.photoimg2)
f_lbl.place(x=1050,y=0,width=520,height=130)
# Top Title Label
title_label = Label(self.root,text="ATTENDANCE MANAGEMENT SYSTEM",font=("times new roman",35,"bold"),bg="white",fg="red")
Line 1, Column 1; Detect Indentation: Setting indentation to 4 spaces
Spaces: 4
Python
ENG
IN
12:31 PM
4/8/2024
C:\Users\Shubham\Desktop\project\Face_Recognition_Attendance_Project\attendance.py (Face_Recognition_Attendance_Project) - Sublime Text (UNREGISTERED)
#-- Function Declaration --
# ****************** Fetching Data From CSV File ******************
def fetch_data(self,rows):
self.Attendance_Report_table.delete(*self.Attendance_Report_table.get_children())
for i in rows:
self.Attendance_Report_table.insert("",END,values=i)
# ****************** Importing CSV File ******************
def import_csv(self):
global my_data , file_name
my_data.clear()
file_name = filedialog.askopenfilename(initialdir=os.getcwd(),title="Open CSV",filetypes=(("CSV File","*.csv"),("ALL File","*.*")),parent=self.root)
with open(file_name) as myfile:
csv_read = csv.reader(myfile,delimiter=",")
headers = next(csv_read) # Skip The First Row
for i in csv_read:
my_data.append(i)
self.fetch_data(my_data)
# ****************** Exporting CSV File ******************
def export_csv(self):
try:
if(len(my_data)) < 1:
messagebox.showerror("Error","No Data Found To Export",parent = self.root)
return False
export_file_name = filedialog.asksaveasfilename(initialdir=os.getcwd(),title="Open CSV",filetypes=(("CSV File","*.csv"),("ALL File","*.*")),parent=self.root)
with open(export_file_name,mode="w",newline="") as myfile:
export_write = csv.writer(myfile,delimiter=",")
for i in my_data:
export_write.writerow(i)
messagebox.showinfo("Success",f"Data Exported To File **{self.export_file_name}** Successfully",parent= self.root)
except Exception as es:
messagebox.showerror("Error",f"Due To :{str(es)}",parent=self.root)
# ****************** Updating CSV File ******************
def update_csv(self):
if self.var_attendance_date.get() == "" or self.var_attendance_status.get() == "Select Status" or self.var_attendance_individual_status.get() == "Select Status" or se
messagebox.showerror("Error","All Fields Are Required",parent=self.root)
else :
Update = messagebox.askyesno("Update","Do You Want To Update This Attendance Record",parent=self.root)
if Update > 0:
selected_item = self.Attendance_Report_table.selection()
if selected_item:
selected_item = selected_item[0]
selected_id = self.Attendance_Report_table.item(selected_item, "values")[0]
data = [] # Load data from the CSV file into memory
with open(file_name, 'r') as csvfile: # Add The CSV Data To Data List
reader = csv.DictReader(csvfile)
for row in reader:
data.append(row)
for row in data: # Update The Data List W.R.T Id Field
if row['ID'] == selected_id:
Line 1, Column 1
Spaces: 4
Python
ENG
IN
12:31 PM
4/8/2024
C:\Users\Shubham\Desktop\project\Face_Recognition_Attendance_Project\attendance.py (Face_Recognition_Attendance_Project) - Sublime Text (UNREGISTERED)
writer.writeheader()
writer.writerows(data)
# Invoice Code After Successfull Completion Of Operation
sound = gtts.gTTS("Attendance Record Has Been Updated Successfully",lang = "en")
sound.save(r"attendance.mp3")
playsound.playsound(r"attendance.mp3")
if os.path.exists("attendance.mp3"):
os.remove("attendance.mp3")
messagebox.showinfo("Success","Attendance Record Has Been Updated Successfully",parent=self.root)
# Populate The Treeview With Updated CSV File
my_data.clear()
with open(file_name) as myfile:
csv_read = csv.reader(myfile,delimiter=",")
headers = next(csv_read) # Skip the first row
for i in csv_read:
my_data.append(i)
self.fetch_data(my_data)
else:
if not Update:
return
# ****************** Reset All The Fields To Default Values ******************
def reset_data(self):
self.var_attendance_status.set("Select Status")
self.var_attendance_id.set("Digits only")
self.var_attendance_date.set("")
self.var_attendance_name.set("")
self.var_attendance_name.set("")
self.var_attendance_individual_status.set("Select Status")
# ****************** Fetch The Data From Table TreeView To The Input Fields ******************
def get_cursor(self,event=""):
cursor_focus = self.Attendance_Report_table.focus()
content = self.Attendance_Report_table.item(cursor_focus)
data = content["values"]
self.var_attendance_id.set(data[0])
self.var_attendance_individual_status.set(data[1])
self.var_attendance_name.set(data[2])
self.var_attendance_date.set(data[3])
self.var_attendance_time.set(data[4])
self.var_attendance_status.set(data[5])
# ****************** Back To Home Function ******************
def back_to_home(self):
self.root.destroy()
# ---------------------------------- Main Class Calling ----------------------------------
if __name__ == "__main__":
root = Tk()
obj = Attendance(root)
root.mainloop()
Line 1, Column 1
Spaces: 4
Python
ENG
IN
12:31 PM
4/8/2024

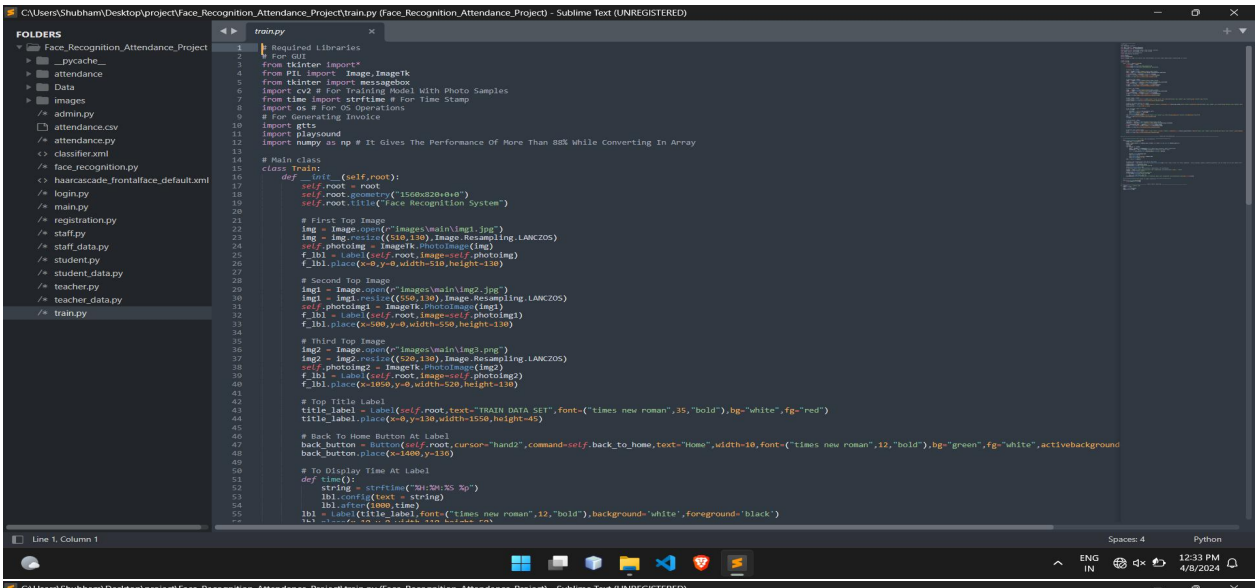


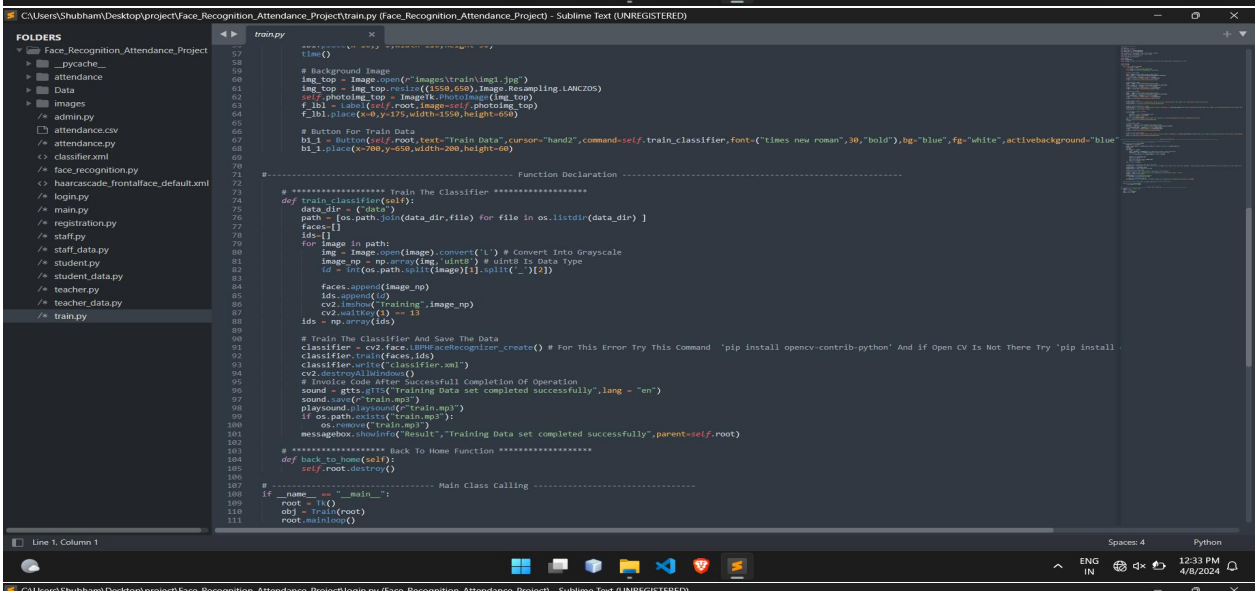

C:\Users\Shubham\Desktop\project\Face_Recognition_Attendance_Project\main.py (Face_Recognition_Attendance_Project) - Sublime Text (UNREGISTERED)
FOLDERS
Face_Recognition_Attendance_Project
__pycache__
attendance
Data
images
admin.py
attendance.csv
attendance.py
classifier.xml
face_recognition.py
haarcascade_frontalface_default.xml
login.py
main.py
registration.py
staff.py
staff_data.py
student.py
student_data.py
teacher.py
teacher_data.py
train.py
main.py
Required Libraries
For GUI
from tkinter import*
from tkinter import ttk
from tkinter import messagebox
from PIL import Image,ImageTk
import os # For OS Operations
from time import strftime # For Time Stamp
# For Including Other Project Windows
from student import Student
from teacher import Teacher_Details
from train import Train
from face_recognition import Face_Recognition
from attendance import Attendance
from staff import Staff_Details
# r Is Used To Convert Back Slash To Forward Slash As In Python We Need To Convert It While Giving Path
# Main class
class Face_Recognition_System :
def __init__(self,root):
self.root = root
self.root.geometry("1560x820+0+0")
self.root.title("Face Recognition System")
# First Top Image
img = Image.open(r"images\main\img1.jpg")
img = img.resize((510,130),Image.Resampling.LANCZOS)
self.photoimg = ImageTk.PhotoImage(img)
f_lbl = Label(self.root,image=self.photoimg)
f_lbl.place(x=0,y=0,width=510,height=130)
# Second Top Image
img1 = Image.open(r"images\main\img2.jpg")
img1 = img1.resize((550,130),Image.Resampling.LANCZOS)
self.photoimg1 = ImageTk.PhotoImage(img1)
f_lbl = Label(self.root,image=self.photoimg1)
f_lbl.place(x=500,y=0,width=550,height=130)
# Third Top Image
img2 = Image.open(r"images\main\img3.png")
img2 = img2.resize((520,130),Image.Resampling.LANCZOS)
self.photoimg2 = ImageTk.PhotoImage(img2)
f_lbl = Label(self.root,image=self.photoimg2)
f_lbl.place(x=1050,y=0,width=520,height=130)
# Background Image
img3 = Image.open(r"images\background.jpg")
img3 = img3.resize((1550,710),Image.Resampling.LANCZOS)
self.photoimg3 = ImageTk.PhotoImage(img3)
bg_img = Label(self.root,image=self.photoimg3)
bg_img.place(x=0,y=130,width=1550,height=710)
# Top Title Label
title_label = Label(bg_img,text="FACE RECOGNITION ATTENDANCE SYSTEM SOFTWARE",font=("times new roman",35,"bold"),bg="white",fg="red")
Line 1, Column 1; Detect Indentation: Setting indentation to 4 spaces
Spaces: 4
Python
ENG
IN
12:32 PM
4/8/2024

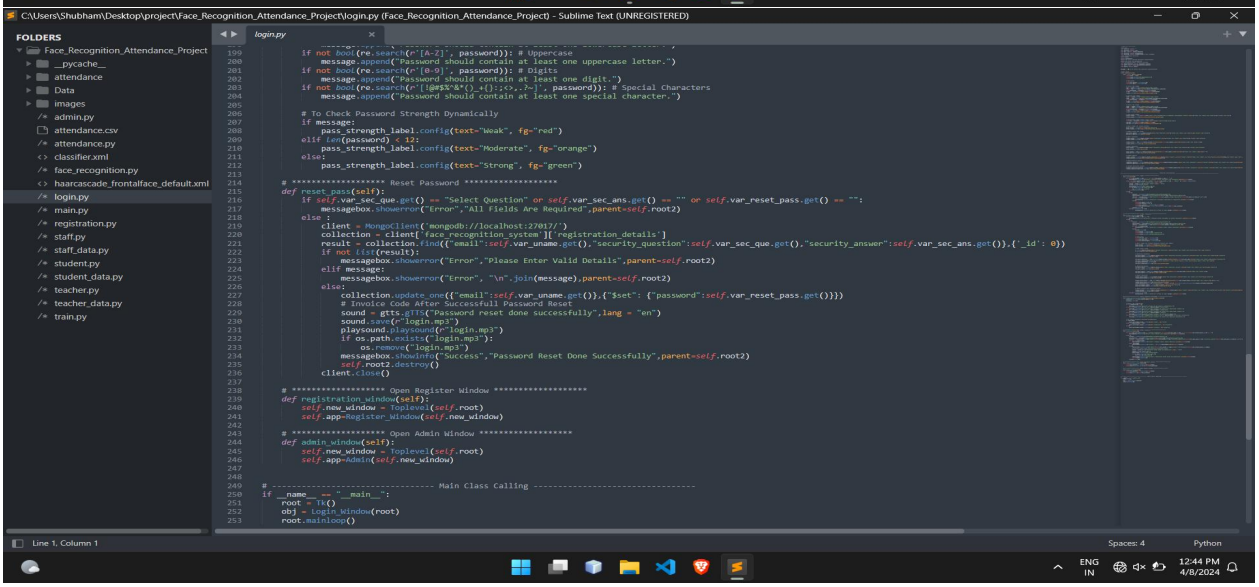

C:\Users\Shubham\Desktop\project\Face_Recognition_Attendance_Project\main.py (Face_Recognition_Attendance_Project) - Sublime Text (UNREGISTERED)
FOLDERS
Face_Recognition_Attendance_Project
main.py
b1.place(x=1100,y=480,width=220,height=220)
b1_1 = Button(bg_img,text="Exit",cursor="hand2",command=self.exit,font=("times new roman",15,"bold"),bg="darkblue",fg="white",activebackground="darkblue",activeforegr
b1_1.place(x=1100,y=580,width=220,height=40)
#---------- Function Declaration ----------
# ********** Student Details Function **********
def student_details(self):
self.new_window = Toplevel(self.root)
self.app=Student(self.new_window)
# ********** Train Data Function **********
def train_data(self):
self.new_window = Toplevel(self.root)
self.app=Train(self.new_window)
# ********** Face Recognition Function **********
def face_data(self):
self.new_window = Toplevel(self.root)
self.app=Face_Recognition(self.new_window)
# ********** Attendance Record Function **********
def attendance_data(self):
self.new_window = Toplevel(self.root)
self.app=Attendance(self.new_window)
# ********** Staff Details Function **********
def staff_details(self):
self.new_window = Toplevel(self.root)
self.app=Staff_Details(self.new_window)
# ********** Teacher Details Function **********
def teacher_details(self):
self.new_window = Toplevel(self.root)
self.app=Teacher_Details(self.new_window)
# ********** Exit From The System Function **********
def exit(self):
self.exit = messagebox.askyesno("Face Recognition","Are you sure",parent=self.root)
if self.exit > 0:
self.root.destroy()
else:
return
# ********** Function To Open 'Data' Folder **********
def open_img(self):
os.startfile("data")
# ---------- Main Class Calling ----------
if __name__ == "__main__":
root = Tk()
obj = Face_Recognition_System(root)
root.mainloop()
Line 1, Column 1
Spaces: 4
Python
ENG
IN
12:32 PM
4/8/2024

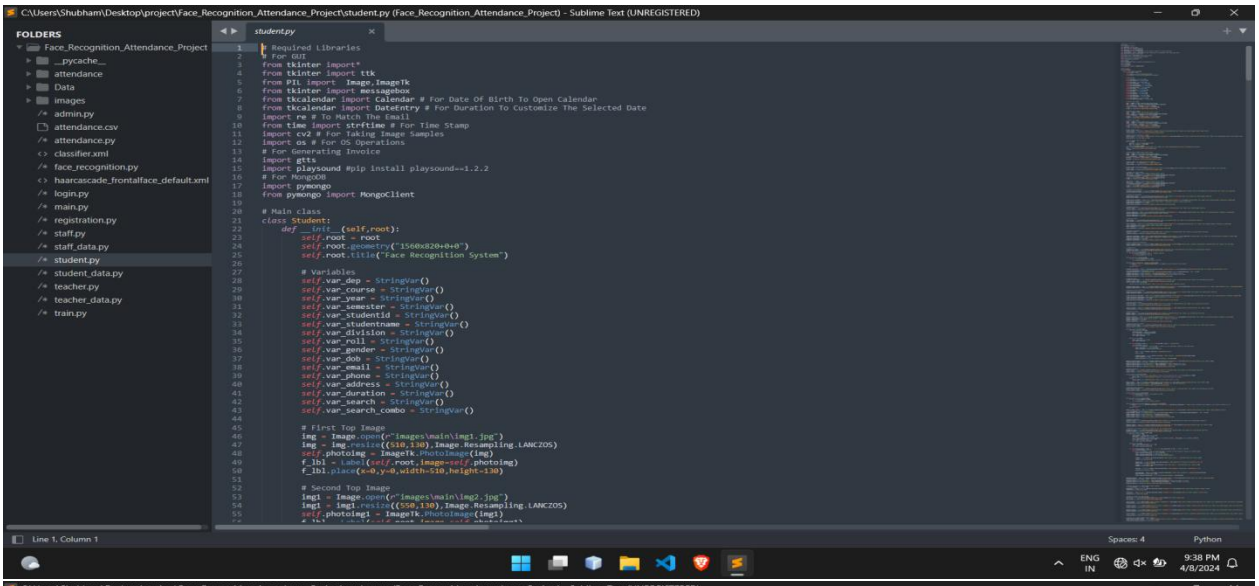

C:\Users\Shubham\Desktop\project\Face_Recognition_Attendance_Project\student.py (Face_Recognition_Attendance_Project) - Sublime Text (UNREGISTERED)
FOLDERS
Face_Recognition_Attendance_Project
student.py
Required Libraries
For GUI
from tkinter import*
from tkinter import ttk
from PIL import Image,ImageTk
from tkinter import messagebox
from tkcalendar import Calendar # For Date Of Birth To Open Calendar
from tkcalendar import DateEntry # For Duration To Customize The Selected Date
import re # To Match The Email
from time import strftime # For Time Stamp
import cv2 # For Taking Image Samples
import os # For OS Operations
# For Generating Invoice
import gtts
import playsound #pip install playsound==1.2.2
# For MongoDB
import pymongo
from pymongo import MongoClient
# Main class
class Student:
def __init__(self,root):
self.root = root
self.root.geometry("1560x820+0+0")
self.root.title("Face Recognition System")
# Variables
self.var_dep = StringVar()
self.var_course = StringVar()
self.var_year = StringVar()
self.var_semester = StringVar()
self.var_studentid = StringVar()
self.var_studentname = StringVar()
self.var_division = StringVar()
self.var_roll = StringVar()
self.var_gender = StringVar()
self.var_dob = StringVar()
self.var_email = StringVar()
self.var_phone = StringVar()
self.var_address = StringVar()
self.var_duration = StringVar()
self.var_search = StringVar()
self.var_search_combo = StringVar()
# First Top Image
img = Image.open(r"images\main\img1.jpg")
img = img.resize((510,130),Image.Resampling.LANCZOS)
self.photoimg = ImageTk.PhotoImage(img)
f_lbl = Label(self.root,image=self.photoimg)
f_lbl.place(x=0,y=0,width=510,height=130)
# Second Top Image
img1 = Image.open(r"images\main\img2.jpg")
img1 = img1.resize((550,130),Image.Resampling.LANCZOS)
self.photoimg1 = ImageTk.PhotoImage(img1)
Line 1, Column 1; Detect Indentation: Setting indentation to 4 spaces
Spaces: 4
Python
ENG
IN
12:32 PM
4/8/2024

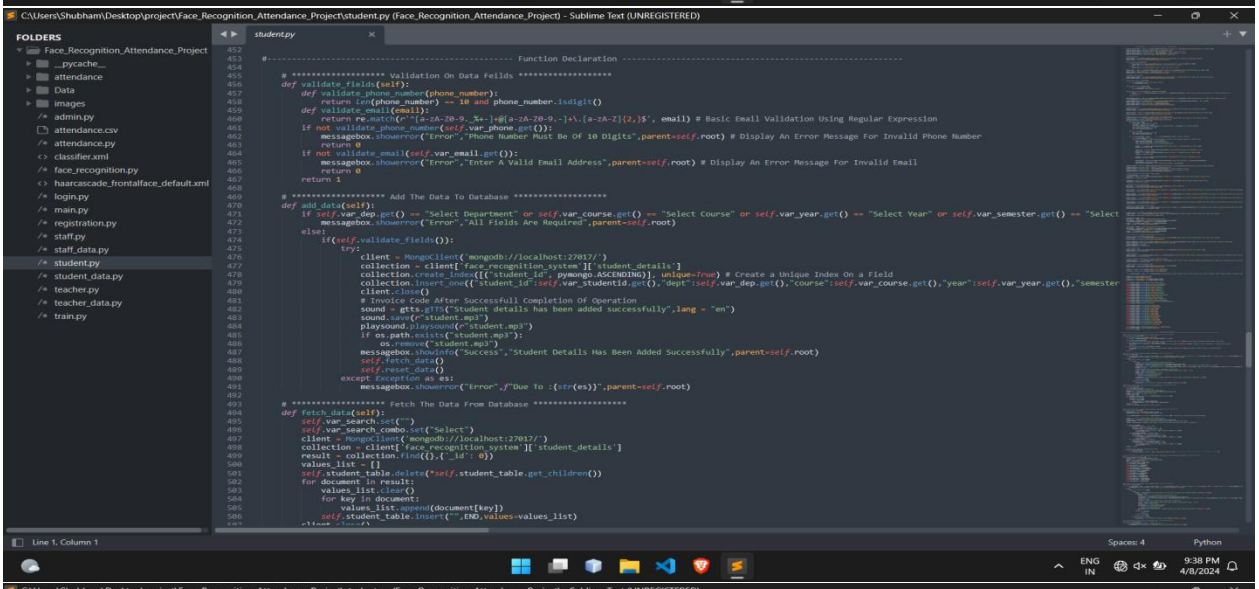


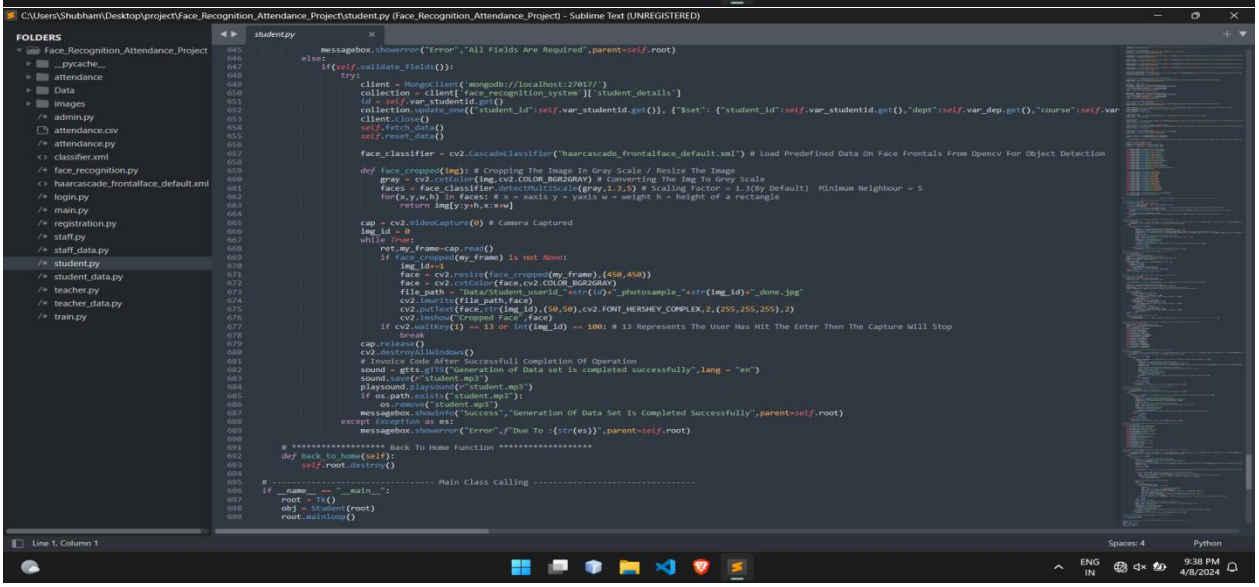

C:\Users\Shubham\Desktop\project\Face_Recognition_Attendance_Project\train.py (Face_Recognition_Attendance_Project) - Sublime Text (UNREGISTERED)
FOLDERS
Face_Recognition_Attendance_Project
__pycache__
attendance
Data
images
admin.py
attendance.csv
attendance.py
classifier.xml
face_recognition.py
haarcascade_frontalface_default.xml
login.py
main.py
registration.py
staff.py
staff_data.py
student.py
student_data.py
teacher.py
teacher_data.py
train.py
train.py
Required Libraries
For GUI
from tkinter import*
from PIL import Image,ImageTk
from tkinter import messagebox
import cv2 # For Training Model With Photo Samples
from time import strftime # For Time Stamp
import os # For OS Operations
# For Generating Invoice
import gtts
import playsound
import numpy as np # It Gives The Performance Of More Than 88% While Converting In Array
# Main Class
class Train:
def __init__(self,root):
self.root = root
self.root.geometry("1560x820+0+0")
self.root.title("Face Recognition System")
# First Top Image
img = Image.open(r"images\main\img1.jpg")
img = img.resize((510,130),Image.Resampling.LANCZOS)
self.photoimg = ImageTk.PhotoImage(img)
f_lbl = Label(self.root,image=self.photoimg)
f_lbl.place(x=0,y=0,width=510,height=130)
# Second Top Image
img1 = Image.open(r"images\main\img2.jpg")
img1 = img1.resize((550,130),Image.Resampling.LANCZOS)
self.photoimg1 = ImageTk.PhotoImage(img1)
f_lbl = Label(self.root,image=self.photoimg1)
f_lbl.place(x=500,y=0,width=550,height=130)
# Third Top Image
img2 = Image.open(r"images\main\img3.png")
img2 = img2.resize((520,130),Image.Resampling.LANCZOS)
self.photoimg2 = ImageTk.PhotoImage(img2)
f_lbl = Label(self.root,image=self.photoimg2)
f_lbl.place(x=1050,y=0,width=520,height=130)
# Top Title Label
title_label = Label(self.root,text="TRAIN DATA SET",font=("times new roman",35,"bold"),bg="white",fg="red")
title_label.place(x=0,y=130,width=1550,height=45)
# Back To Home Button At Label
back_button = Button(self.root,cursor="hand2",command=self.back_to_home,text="Home",width=10,font=("times new roman",12,"bold"),bg="green",fg="white",activebackground
back_button.place(x=1400,y=130)
# To Display Time At Label
def time():
string = strftime("%H:%M:%S %p")
lbl.config(text = string)
lbl.after(1000,time)
lbl = Label(title_label,font=("times new roman",12,"bold"),background='white',foreground='black')
Line 1, Column 1
Spaces: 4
Python
ENG
IN
12:33 PM
4/8/2024

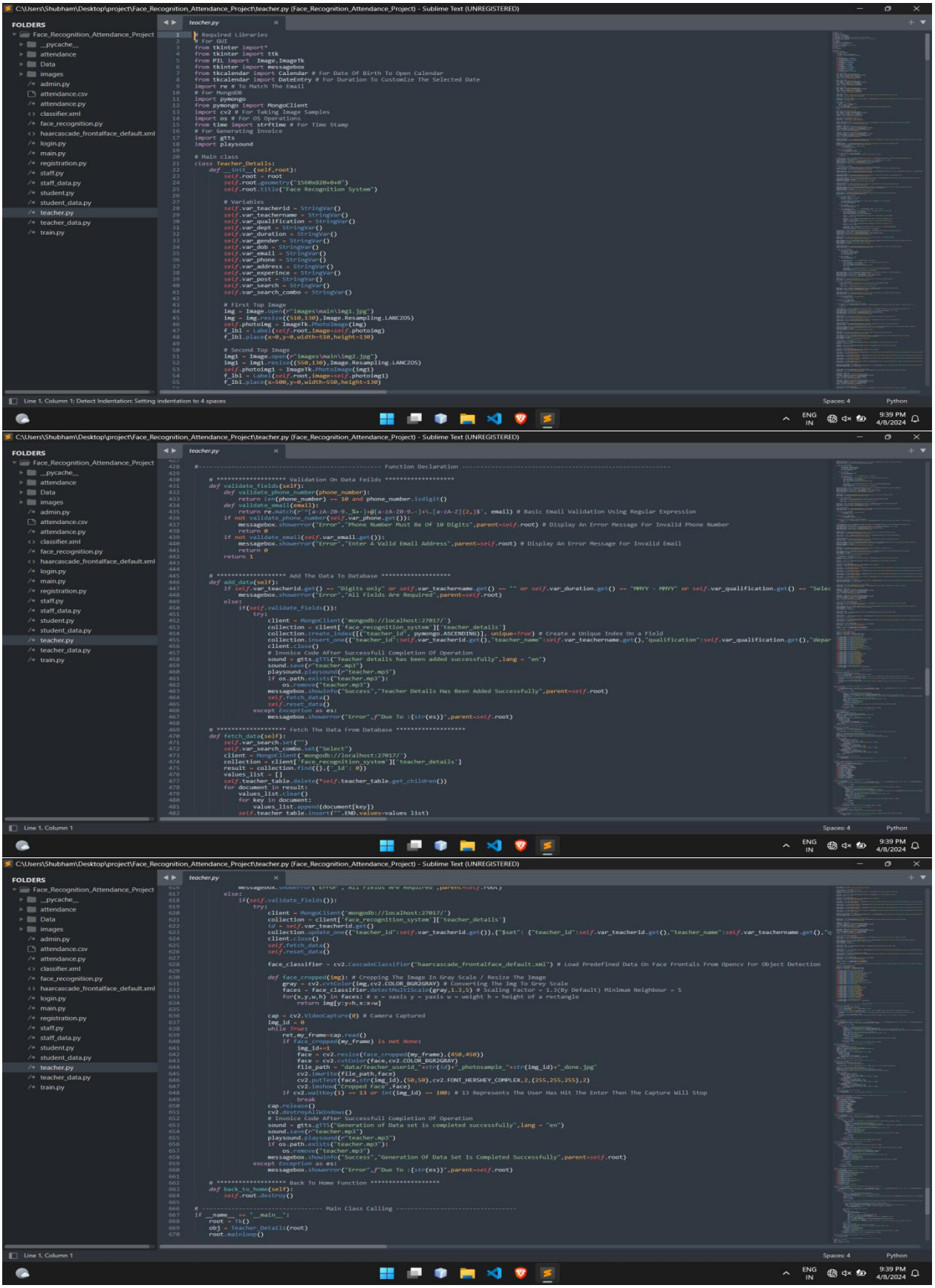

C:\Users\Shubham\Desktop\project\Face_Recognition_Attendance_Project\train.py (Face_Recognition_Attendance_Project) - Sublime Text (UNREGISTERED)
FOLDERS
Face_Recognition_Attendance_Project
__pycache__
attendance
Data
images
admin.py
attendance.csv
attendance.py
classifier.xml
face_recognition.py
haarcascade_frontalface_default.xml
login.py
main.py
registration.py
staff.py
staff_data.py
student.py
student_data.py
teacher.py
teacher_data.py
train.py
train.py
time()
# Background Image
img_top = Image.open(r"images\train\img1.jpg")
img_top = img_top.resize((1550,650),Image.Resampling.LANCZOS)
self.photoimg_top = ImageTk.PhotoImage(img_top)
f_lbl = Label(self.root,image=self.photoimg_top)
f_lbl.place(x=0,y=175,width=1550,height=650)
# Button For Train Data
b1_1 = Button(self.root,text="Train Data",cursor="hand2",command=self.train_classifier,font=("times new roman",30,"bold"),bg="blue",fg="white",activebackground="blue"
b1_1.place(x=700,y=650,width=200,height=60)
#-- Function Declaration --
# ****************** Train The Classifier ******************
def train_classifier(self):
data_dir = ("data")
path = [os.path.join(data_dir,file) for file in os.listdir(data_dir)]
faces=[]
ids=[]
for image in path:
img = Image.open(image).convert('L') # Convert Into Grayscale
image_np = np.array(img,'uint8') # uint8 Is Data Type
id = int(os.path.split(image)[1].split('_')[2])
faces.append(image_np)
ids.append(id)
cv2.imshow("Training",image_np)
cv2.waitKey(1) == 13
ids = np.array(ids)
# Train The Classifier And Save The Data
classifier = cv2.face.LBPHFaceRecognizer_create() # For This Error Try This Command 'pip install opencv-contrib-python' And if Open CV Is Not There Try 'pip install
classifier.train(faces,ids)
classifier.write("classifier.xml")
cv2.destroyAllWindows()
# Invoice Code After Successfull Completion Of Operation
sound = gtts.gTTS("Training Data set completed successfully",lang = "en")
sound.save(r"train.mp3")
playsound.playsound(r"train.mp3")
if os.path.exists("train.mp3"):
os.remove("train.mp3")
messagebox.showinfo("Result","Training Data set completed successfully",parent=self.root)
# ****************** Back To Home Function ******************
def back_to_home(self):
self.root.destroy()
# --------------------------------- Main Class Calling ---------------------------------
if __name__ == "__main__":
root = Tk()
obj = Train(root)
root.mainloop()
Line 1, Column 1
Spaces: 4
Python
ENG
IN
12:33 PM
4/8/2024


C:\Users\Shubham\Desktop\project\Face_Recognition_Attendance_Project\login.py (Face_Recognition_Attendance_Project) - Sublime Text (UNREGISTERED)
FOLDERS
Face_Recognition_Attendance_Project
__pycache__
attendance
Data
images
admin.py
attendance.csv
attendance.py
classifier.xml
face_recognition.py
haarcascade_frontalface_default.xml
login.py
main.py
registration.py
staff.py
staff_data.py
student.py
student_data.py
teacher.py
teacher_data.py
train.py
login.py
if not bool(re.search(r'[A-Z]', password)): # Uppercase
message.append("Password should contain at least one uppercase letter.")
if not bool(re.search(r'[0-9]', password)): # Digits
message.append("Password should contain at least one digit.")
if not bool(re.search(r'[@#$%^&*()_+{};:<?,./~]', password)): # Special Characters
message.append("Password should contain at least one special character.")
# To Check Password Strength Dynamically
if message:
pass_strength_label.config(text="Weak", fg="red")
elif len(password) < 12:
pass_strength_label.config(text="Moderate", fg="orange")
else:
pass_strength_label.config(text="Strong", fg="green")
# ****************** Reset Password ******************
def reset_pass(self):
if self.var_sec_que.get() == "Select Question" or self.var_sec_ans.get() == "" or self.var_reset_pass.get() == "":
messagebox.showerror("Error","All Fields Are Required",parent=self.root2)
else:
client = MongoClient('mongodb://localhost:27017/')
collection = client['face_recognition_system']['registration_details']
result = collection.find({"email":self.var_uname.get(),"security_question":self.var_sec_que.get(),"security_answer":self.var_sec_ans.get()},{'_id': 0})
if not list(result):
messagebox.showerror("Error","Please Enter Valid Details",parent=self.root2)
elif message:
messagebox.showerror("Error", "\n".join(message),parent=self.root2)
else:
collection.update_one({"email":self.var_uname.get()},{"$set": {"password":self.var_reset_pass.get()}})
# Invoice Code After Successfull Password Reset
sound = gtts.gTTS("Password reset done successfully",lang = "en")
sound.save(r"login.mp3")
playsound.playsound(r"login.mp3")
if os.path.exists("login.mp3"):
os.remove("login.mp3")
messagebox.showinfo("Success","Password Reset Done Successfully",parent=self.root2)
self.root2.destroy()
client.close()
# ****************** Open Register Window ******************
def registration_window(self):
self.new_window = Toplevel(self.root)
self.app=Register_Window(self.new_window)
# ****************** Open Admin Window ******************
def admin_window(self):
self.new_window = Toplevel(self.root)
self.app=Admin(self.new_window)
# --------------------------------- Main Class Calling ---------------------------------
if __name__ == "__main__":
root = Tk()
obj = Login_Window(root)
root.mainloop()
Line 1, Column 1
Spaces: 4
Python
ENG
IN
12:44 PM
4/8/2024

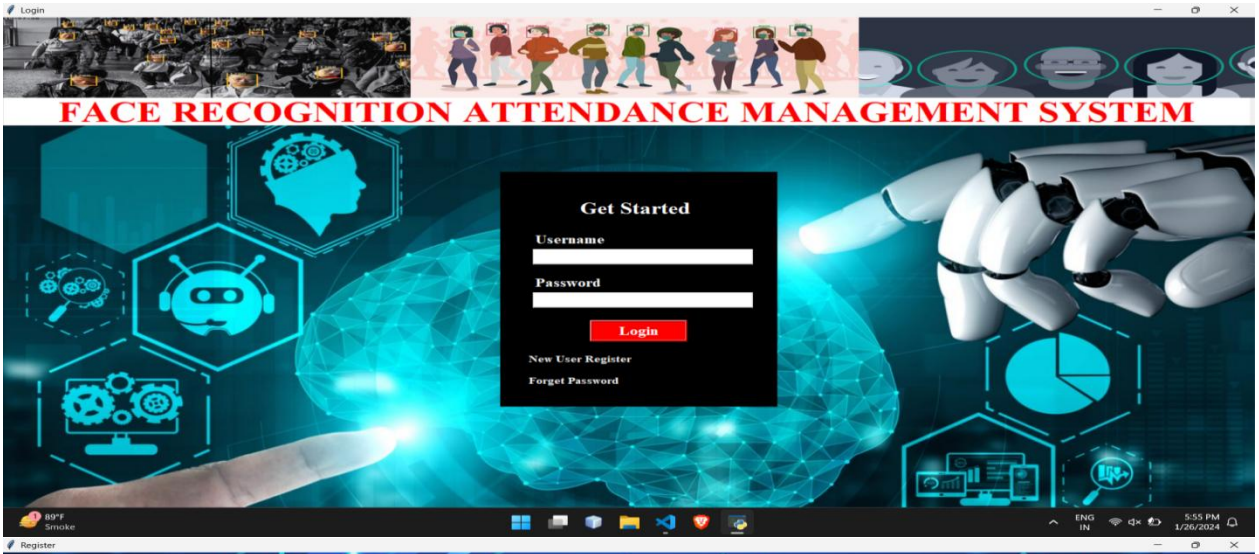

C:\Users\Shubham\Desktop\project\Face_Recognition_Attendance_Project\student.py (Face_Recognition_Attendance_Project) - Sublime Text (UNREGISTERED)
FOLDERS
Face_Recognition_Attendance_Project
__pycache__
attendance
Data
images
admin.py
attendance.csv
attendance.py
classifier.xml
face_recognition.py
haarcascade_frontalface_default.xml
login.py
main.py
registration.py
staff.py
staff_data.py
student.py
student_data.py
teacher.py
teacher_data.py
train.py
student.py
Required Libraries
For GUI
from tkinter import*
from tkinter import ttk
from PIL import Image,ImageTk
from tkinter import messagebox
from tkcalendar import Calendar # For Date Of Birth To Open Calendar
from tkcalendar import DateEntry # For Duration To Customize The Selected Date
import re # To Match The Email
from time import strftime # For Time Stamp
import cv2 # For Taking Image Samples
import os # For OS Operations
# For Generating Invoice
import gtts
import playsound #pip install playsound==1.2.2
# For MongoDB
import pymongo
from pymongo import MongoClient
# Main class
class Student:
def __init__(self,root):
self.root = root
self.root.geometry("1560x820+0+0")
self.root.title("Face Recognition System")
# Variables
self.var_dep = StringVar()
self.var_course = StringVar()
self.var_year = StringVar()
self.var_semester = StringVar()
self.var_studentId = StringVar()
self.var_studentname = StringVar()
self.var_division = StringVar()
self.var_roll = StringVar()
self.var_gender = StringVar()
self.var_dob = StringVar()
self.var_email = StringVar()
self.var_phone = StringVar()
self.var_address = StringVar()
self.var_duration = StringVar()
self.var_search = StringVar()
self.var_search_Combo = StringVar()
# First Top Image
img = Image.open(r"images\main\img1.jpg")
img = img.resize((510,130),Image.Resampling.LANCZOS)
self.photoimg = ImageTk.PhotoImage(img)
f_lbl = Label(self.root,image=self.photoimg)
f_lbl.place(x=0,y=0,width=510,height=130)
# Second Top Image
img1 = Image.open(r"images\main\img2.jpg")
img1 = img1.resize((550,130),Image.Resampling.LANCZOS)
self.photoimg1 = ImageTk.PhotoImage(img1)
Line 1, Column 1
Spaces: 4
Python
ENG
IN
9:38 PM
4/8/2024

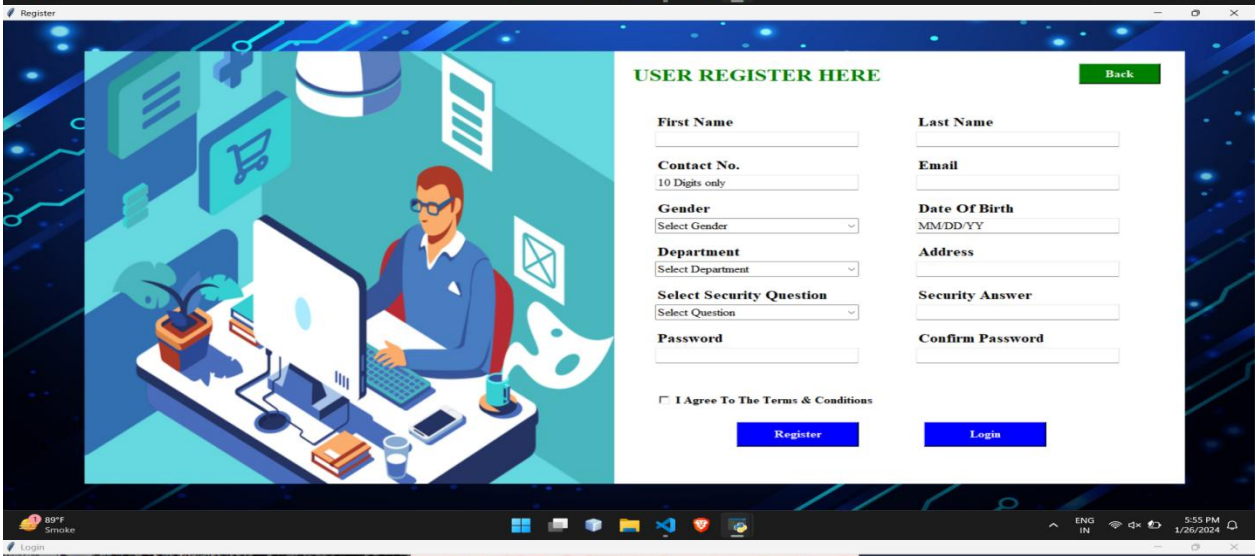

C:\Users\Shubham\Desktop\project\Face_Recognition_Attendance_Project\student.py (Face_Recognition_Attendance_Project) - Sublime Text (UNREGISTERED)
FOLDERS
Face_Recognition_Attendance_Project
__pycache__
attendance
Data
images
admin.py
attendance.csv
attendance.py
classifier.xml
face_recognition.py
haarcascade_frontalface_default.xml
login.py
main.py
registration.py
staff.py
staff_data.py
student.py
student_data.py
teacher.py
teacher_data.py
train.py
student.py
#---------- Function Declaration ----------
# ****************** Validation On Data Feilds ******************
def validate_fields(self):
def validate_phone_number(phone_number):
return len(phone_number) == 10 and phone_number.isdigit()
def validate_email(email):
return re.match(r"^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$", email) # Basic Email Validation Using Regular Expression
if not validate_phone_number(self.var_phone.get()):
messagebox.showerror("Error","Phone Number Must Be Of 10 Digits",parent=self.root) # Display An Error Message For Invalid Phone Number
return 0
if not validate_email(self.var_email.get()):
messagebox.showerror("Error","Enter A Valid Email Address",parent=self.root) # Display An Error Message For Invalid Email
return 0
return 1
# ****************** Add The Data To Database ******************
def add_data(self):
if self.var_dep.get() == "Select Department" or self.var_course.get() == "Select Course" or self.var_year.get() == "Select Year" or self.var_semester.get() == "Select
messagebox.showerror("Error","All Fields Are Required",parent=self.root)
else:
if(self.validate_fields()):
try:
client = MongoClient('mongodb://localhost:27017/')
collection = client['face_recognition_system']['student_details']
collection.create_index([("student_id", pymongo.ASCENDING)], unique=True) # Create a Unique Index On a Field
collection.insert_one({"student_id":self.var_studentId.get(),"dept":self.var_dep.get(),"course":self.var_course.get(),"year":self.var_year.get(),"semester
client.close()
# Invoice Code After Successfull Completion Of Operation
sound = gtts.gTTS("Student details has been added successfully",lang = "en")
sound.save(r"student.mp3")
playsound.playsound(r"student.mp3")
if os.path.exists("student.mp3"):
os.remove("student.mp3")
messagebox.showinfo("Success","Student Details Has Been Added Successfully",parent=self.root)
self.fetch_data()
self.reset_data()
except Exception as es:
messagebox.showerror("Error",f"Due To :{str(es)}",parent=self.root)
# ****************** Fetch The Data From Database ******************
def fetch_data(self):
self.var_search.set("")
self.var_search_Combo.set("Select")
client = MongoClient('mongodb://localhost:27017/')
collection = client['face_recognition_system']['student_details']
result = collection.find({},{"_id": 0})
values_list = []
self.student_table.delete(*self.student_table.get_children())
for document in result:
values_list.clear()
for key in document:
values_list.append(document[key])
self.student_table.insert("",END,values=values_list)
Line 1, Column 1
Spaces: 4
Python
ENG
IN
9:38 PM
4/8/2024

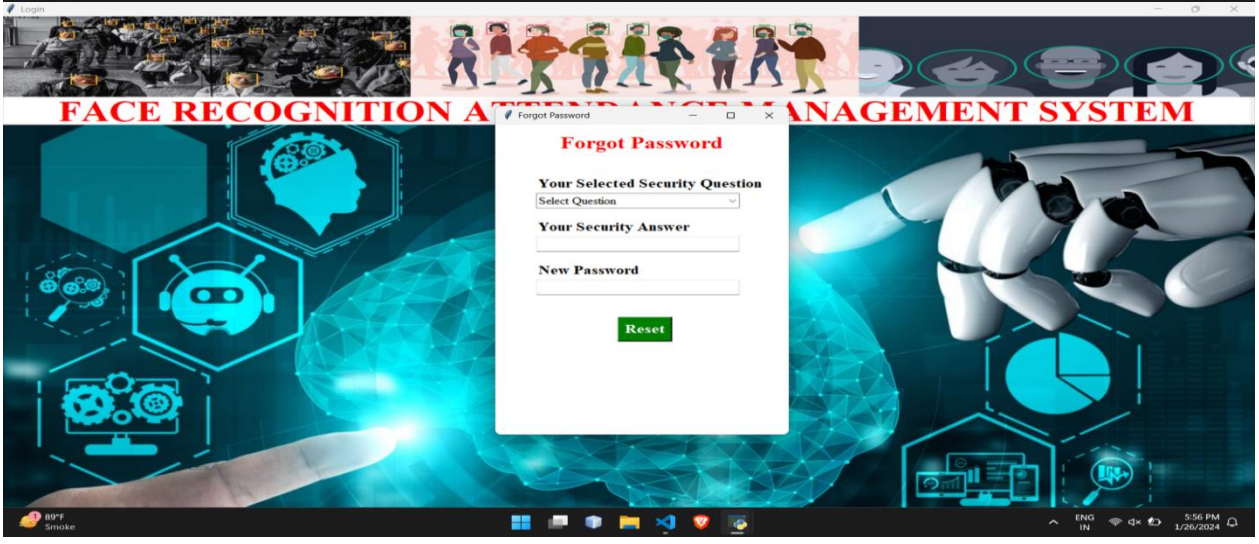

C:\Users\Shubham\Desktop\project\Face_Recognition_Attendance_Project\student.py (Face_Recognition_Attendance_Project) - Sublime Text (UNREGISTERED)
FOLDERS
Face_Recognition_Attendance_Project
__pycache__
attendance
Data
images
admin.py
attendance.csv
attendance.py
classifier.xml
face_recognition.py
haarcascade_frontalface_default.xml
login.py
main.py
registration.py
staff.py
staff_data.py
student.py
student_data.py
teacher.py
teacher_data.py
train.py
student.py
messagebox.showerror("Error","All Fields Are Required",parent=self.root)
else:
if(self.validate_fields()):
try:
client = MongoClient('mongodb://localhost:27017/')
collection = client['face_recognition_system']['student_details']
id = self.var_studentId.get()
collection.update_one({"student_id":self.var_studentId.get()}, {"$set": {"student_id":self.var_studentId.get(),"dept":self.var_dep.get(),"course":self.var
client.close()
self.fetch_data()
self.reset_data()
face_classifier = cv2.CascadeClassifier("haarcascade_frontalface_default.xml") # Load Predefined Data On Face Frontals From Opencv For Object Detection
def face_cropped(img): # Cropping The Image In Gray Scale / Resize The Image
gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY) # Converting The Img To Grey Scale
faces = face_classifier.detectMultiScale(gray,1.3,5) # Scaling Factor = 1.3(By Default) Minimum Neighbour = 5
for(x,y,w,h) in faces: # x - xaxis y - yaxis w - weight h - height of a rectangle
return img[y:y+h,x:x+w]
cap = cv2.VideoCapture(0) # Camera Captured
img_id = 0
while True:
ret,my_frame=cap.read()
if face_cropped(my_frame) is not None:
img_id+=1
face = cv2.resize(face_cropped(my_frame),(450,450))
face = cv2.cvtColor(face,cv2.COLOR_BGR2GRAY)
file_path = "Data/Student_userid_"+str(id)+"_photosample_"+str(img_id)+"_done.jpg"
cv2.imwrite(file_path,face)
cv2.putText(face,str(img_id),(50,50),cv2.FONT_HERSHEY_COMPLEX,2,(255,255,255),2)
cv2.imshow("Cropped Face",face)
if cv2.waitKey(1) == 13 or int(img_id) == 100: # 13 Represents The User Has Hit The Enter Then The Capture Will Stop
break
cap.release()
cv2.destroyAllWindows()
# Invoice Code After Successfull Completion Of Operation
sound = gtts.gTTS("Generation of Data set is completed successfully",lang = "en")
sound.save(r"student.mp3")
playsound.playsound(r"student.mp3")
if os.path.exists("student.mp3"):
os.remove("student.mp3")
messagebox.showinfo("Success","Generation Of Data Set Is Completed Successfully",parent=self.root)
except Exception as es:
messagebox.showerror("Error",f"Due To :{str(es)}",parent=self.root)
# ****************** Back To Home Function ******************
def back_to_home(self):
self.root.destroy()
# ---------- Main Class Calling ----------
if __name__ == "__main__":
root = Tk()
obj = Student(root)
root.mainloop()
Line 1, Column 1
Spaces: 4
Python
ENG
IN
9:38 PM
4/8/2024

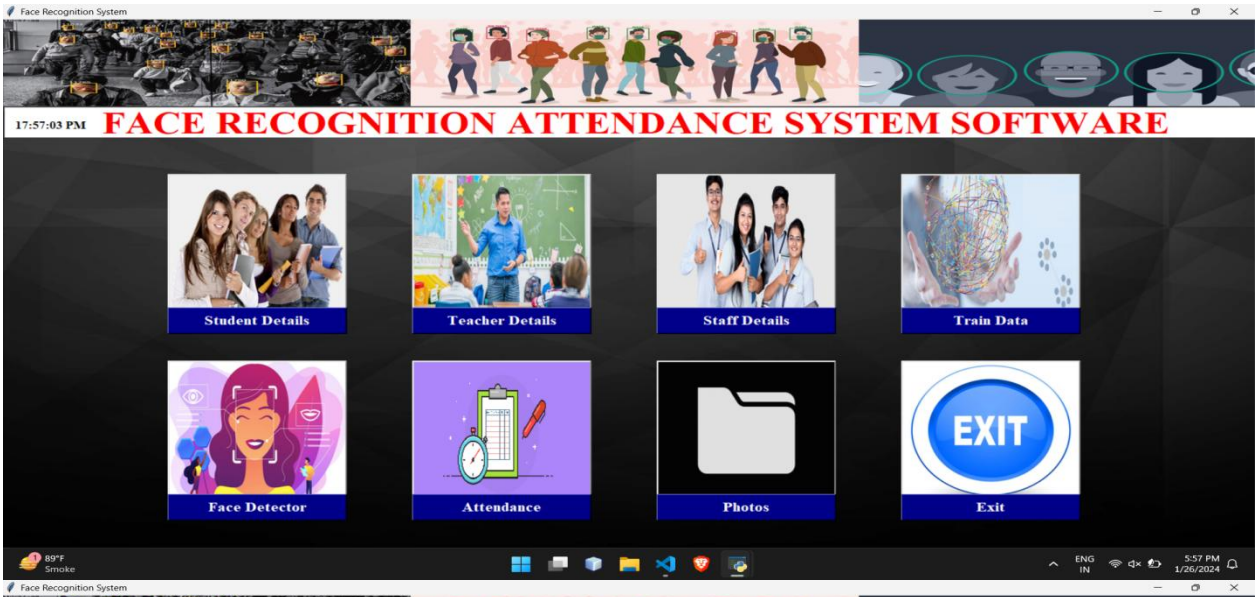


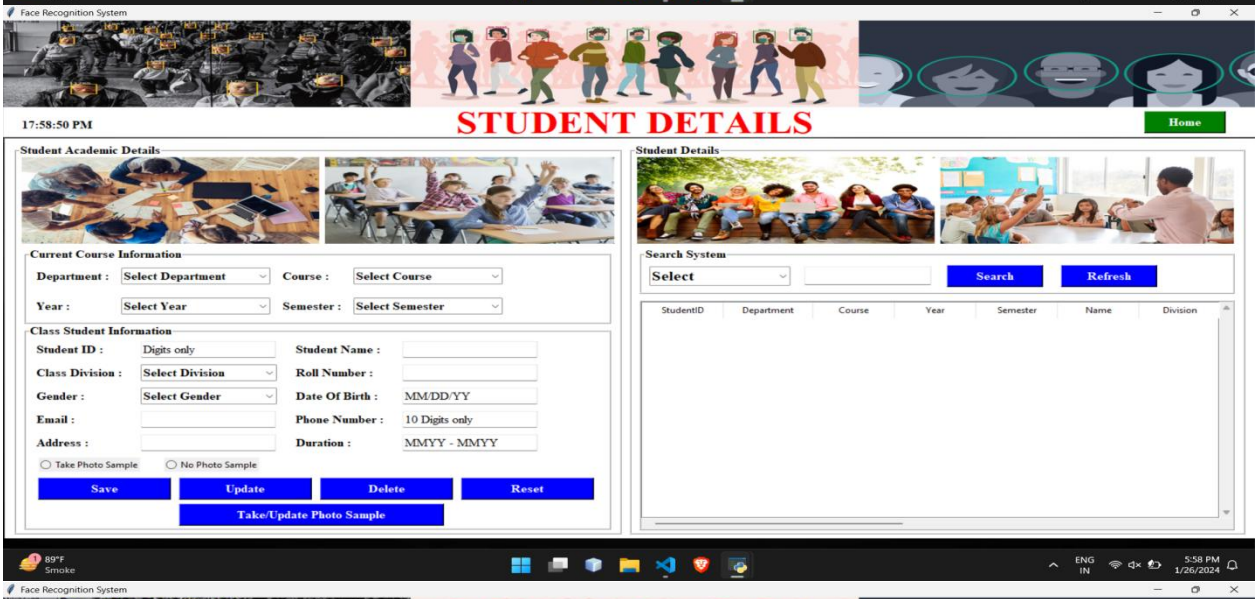
System Screenshots

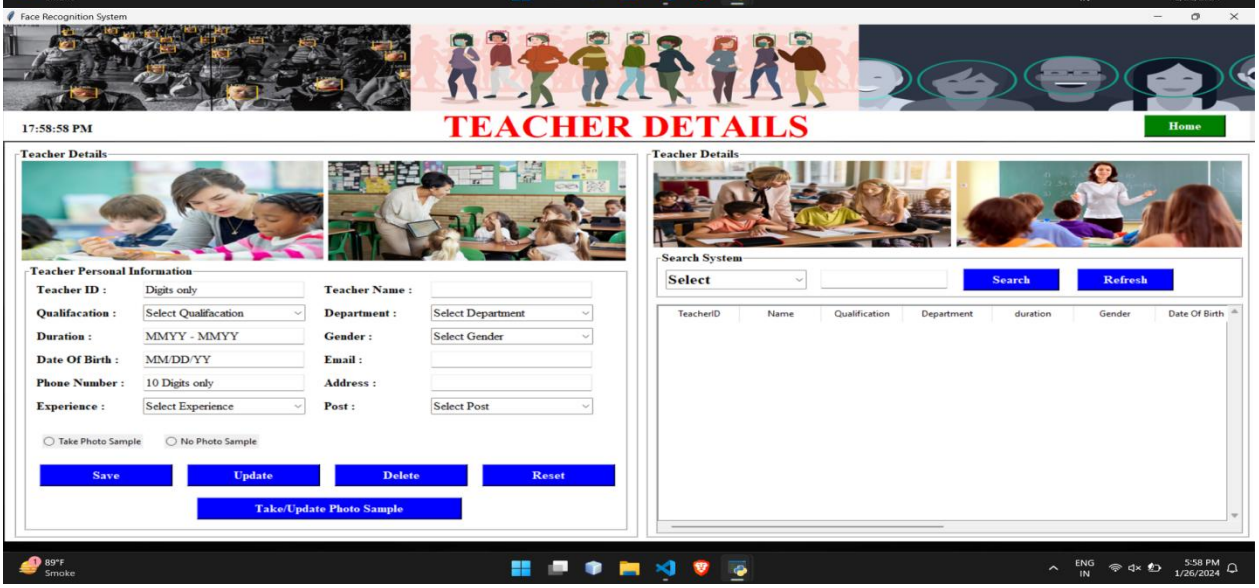


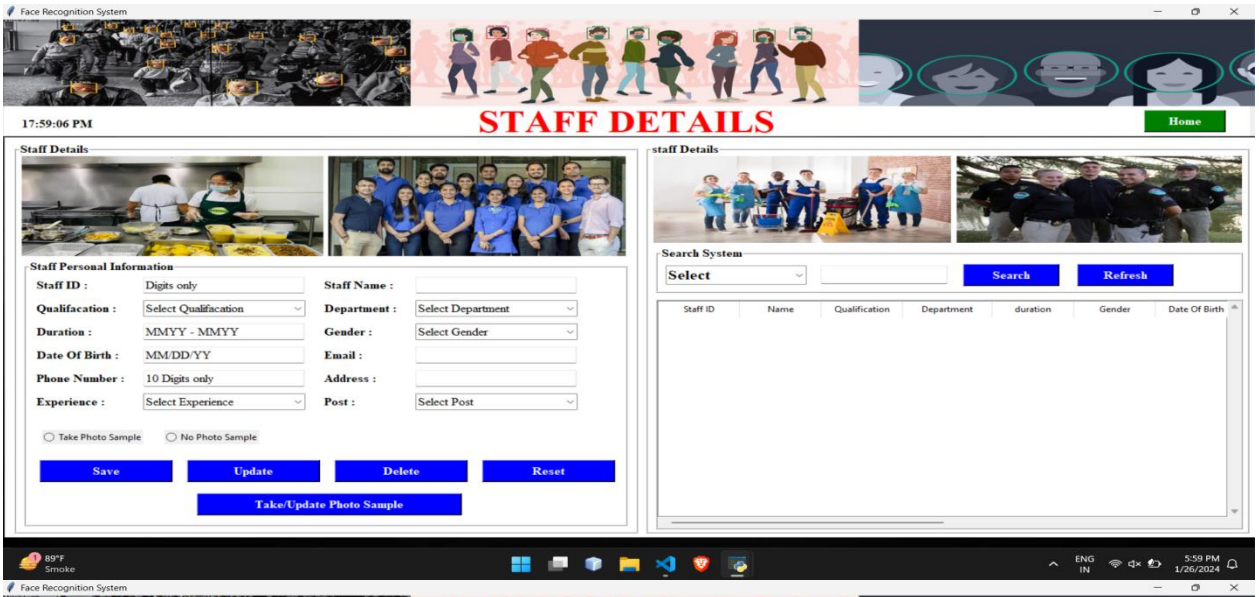


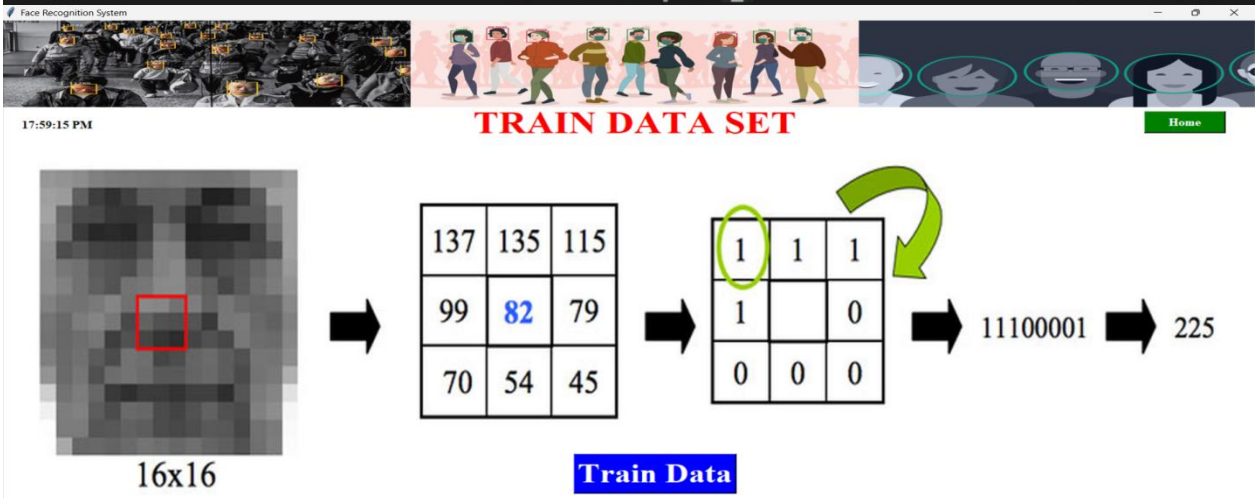


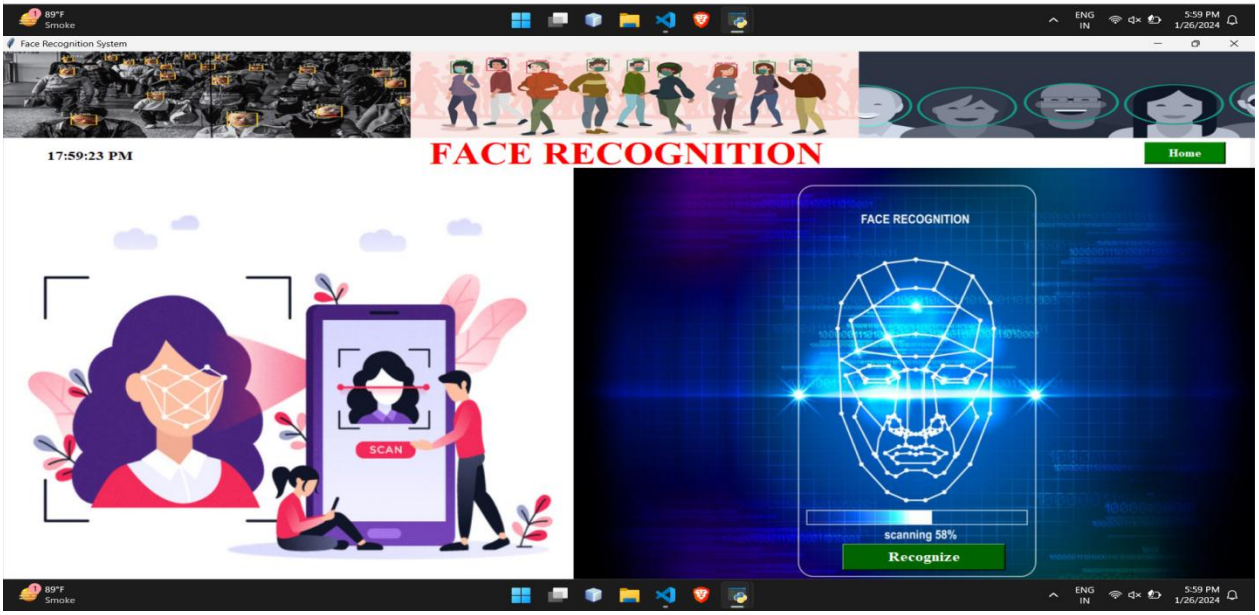

Face Recognition System
17:57:03 PM
FACE RECOGNITION ATTENDANCE SYSTEM SOFTWARE
Student Details
Teacher Details
Staff Details
Train Data
Face Detector
Attendance
Photos
EXIT
Exit

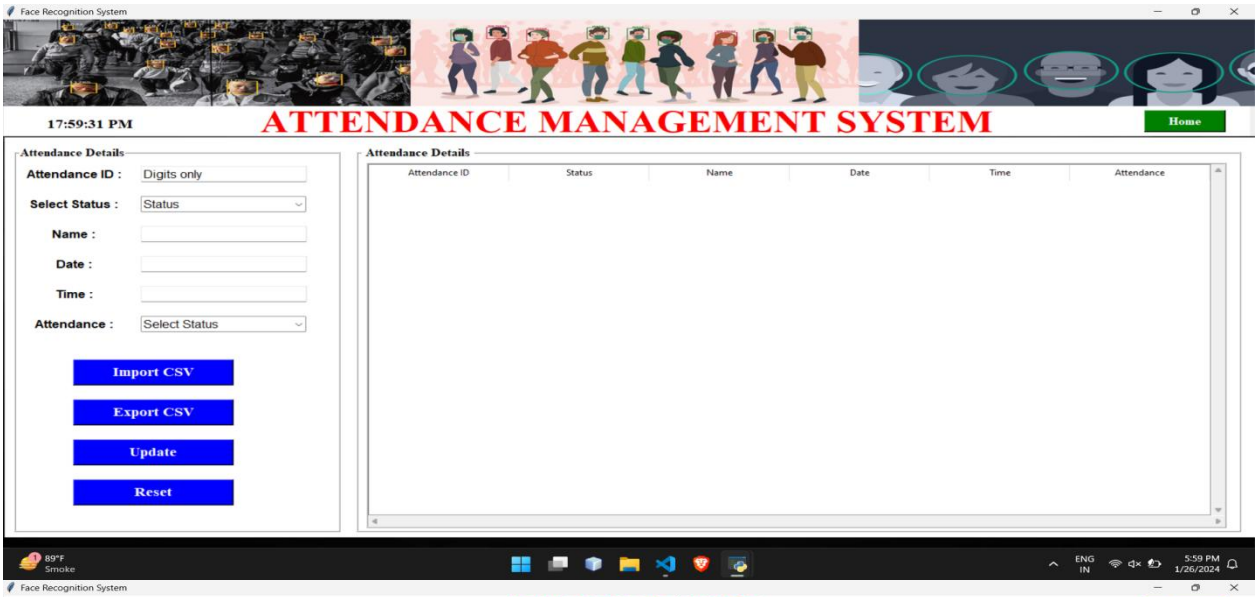

Face Recognition System
17:58:50 PM
STUDENT DETAILS
Home
Student Academic Details
Current Course Information
Department :
Select Department
Course :
Select Course
Year :
Select Year
Semester :
Select Semester
Class Student Information
Student ID :
Digits only
Student Name :
Class Division :
Select Division
Roll Number :
Gender :
Select Gender
Date Of Birth :
MM/DD/YY
Email :
Phone Number :
10 Digits only
Address :
Duration :
MMYY - MMYY
Take Photo Sample
No Photo Sample
Save
Update
Delete
Reset
Take/Update Photo Sample
Student Details
Search System
Select
Search
Refresh
StudentID
Department
Course
Year
Semester
Name
Division

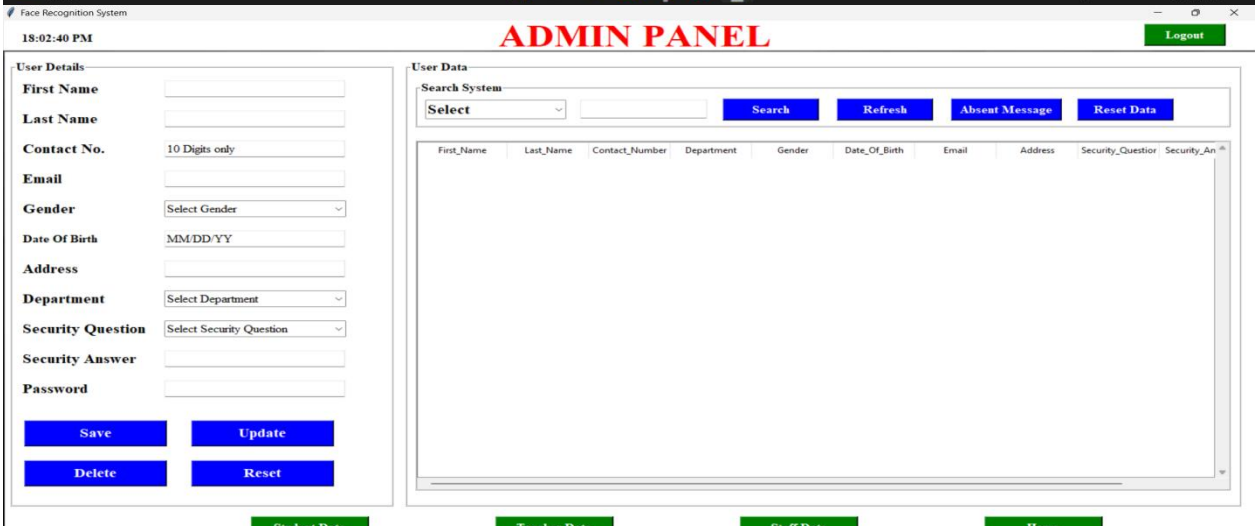

Face Recognition System
17:58:58 PM
TEACHER DETAILS
Home
Teacher Details
Teacher Personal Information
Teacher ID :
Digits only
Teacher Name :
Qualifacation :
Select Qualifacation
Department :
Select Department
Duration :
MMYY - MMYY
Gender :
Select Gender
Date Of Birth :
MM/DD/YY
Email :
Phone Number :
10 Digits only
Address :
Experience :
Select Experience
Post :
Select Post
Take Photo Sample
No Photo Sample
Save
Update
Delete
Reset
Take/Update Photo Sample
Teacher Details
Search System
Select
Search
Refresh
TeacherID
Name
Qualification
Department
duration
Gender
Date Of Birth

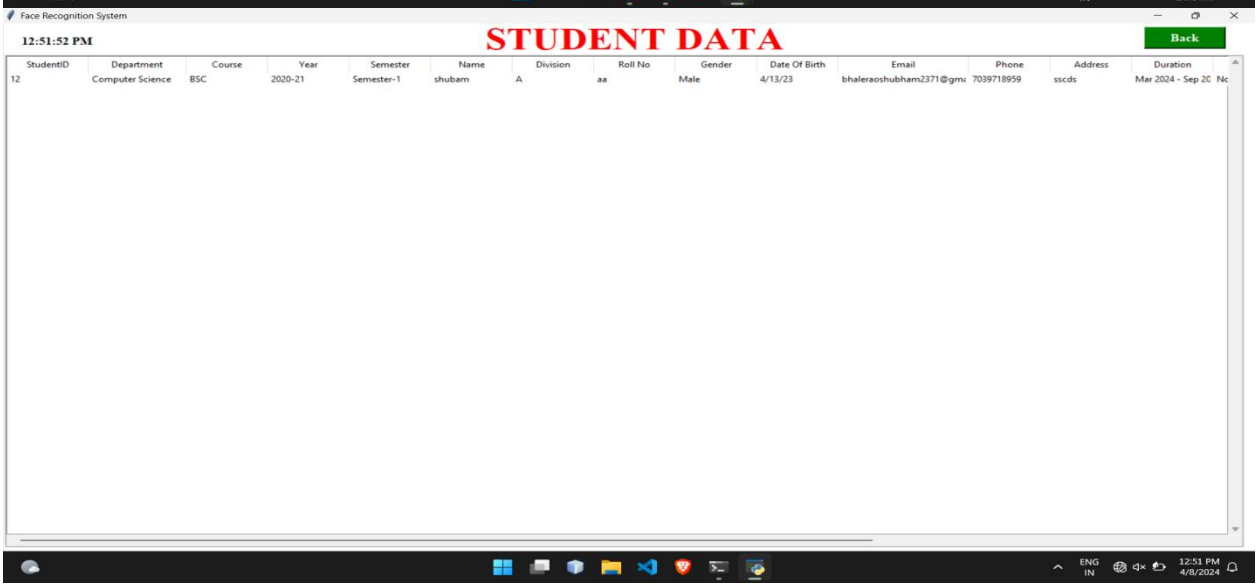

Face Recognition System
17:59:06 PM
STAFF DETAILS
Home
Staff Details
Staff Personal Information
Staff ID :
Digits only
Staff Name :
Qualifacation :
Select Qualifacation
Department :
Select Department
Duration :
MMYY - MMYY
Gender :
Select Gender
Date Of Birth :
MM/DD/YY
Email :
Phone Number :
10 Digits only
Address :
Experience :
Select Experience
Post :
Select Post
Take Photo Sample
No Photo Sample
Save
Update
Delete
Reset
Take/Update Photo Sample
staff Details
Search System
Select
Search
Refresh
Staff ID
Name
Qualification
Department
duration
Gender
Date Of Birth


Face Recognition System
17:59:15 PM
TRAIN DATA SET
Home
16x16
137 135 115
99 82 79
70 54 45
1 1 1
1 0
0 0 0
11100001
225
Train Data


Face Recognition System
17:59:23 PM
FACE RECOGNITION
Home
SCAN
FACE RECOGNITION
scanning 58%
Recognize


Face Recognition System
17:59:31 PM
ATTENDANCE MANAGEMENT SYSTEM
Home
Attendance Details
Attendance ID :
Digits only
Select Status :
Status
Name :
Date :
Time :
Attendance :
Select Status
Import CSV
Export CSV
Update
Reset
Attendance Details
Attendance ID
Status
Name
Date
Time
Attendance


Face Recognition System
18:02:40 PM
ADMIN PANEL
Logout
User Details
First Name
Last Name
Contact No.
10 Digits only
Email
Gender
Select Gender
Date Of Birth
MM/DD/YY
Address
Department
Select Department
Security Question
Select Security Question
Security Answer
Password
Save
Update
Delete
Reset
User Data
Search System
Select
Search
Refresh
Absent Message
Reset Data
First_Name
Last_Name
Contact_Number
Department
Gender
Date_Of_Birth
Email
Address
Student Data
Teacher Data
Staff Data
Home


Face Recognition System
12:51:52 PM
STUDENT DATA
Back
StudentID
Department
Course
Year
Semester
Name
Division
Roll No
Gender
Date Of Birth
Email
Phone
Address
Duration
12
Computer Science
BSC
2020-21
Semester-1
shubam
A
aa
Male
4/13/23
7039718959
sscds
Mar 2024 - Sep 20

Scope for Future Enhancement

I plan to improve the model by incorporating advancements in machine learning and data analysis. Certainly! FaceTrek Attendance - Face recognition Attendance System have a wide range of applications and continuous enhancements can lead to further improvements in performance, accuracy, and usability. I will attempt to incorporate the following future scope areas for improving FaceTrek Attendance.

- **Improved Accuracy and Reliability:** My goal is to refine the algorithm to make face recognition more accurate in difficult situations including dim lighting, occlusion, changing expressions on the face, and changing postures.
- **Privacy and Security:** To guarantee that facial data is safely maintained and utilized only for approved purposes, I will create strong privacy-preserving methods. Incorporate cutting-edge encryption techniques to safeguard facial data both in transit and storage.
- **Real-time Processing:** My goal is to enhance hardware and algorithms for real-time face identification in high-definition video streams. Creating effective parallel processing strategies to manage complex, real-time face recognition workloads.
- **Ethical Considerations:** I will discuss the moral issues of bias, equity, and transparency in FaceTrek Attendance, with a focus on potential abuse and demographic inequalities. Establish auditing and accountability procedures to guarantee that FaceTrek Attendance is utilized in an ethical and responsible manner.
- **Adaptability and Robustness:** I'll create robust algorithms that are able to learn and adjust over time, accounting for things like changing hairstyles, aging-related changes in face appearance, and more. By utilizing adversarial training and anti-spoofing strategies, I can improve resilience against spoofing attacks and manipulations.
- **User Experience:** By creating user-friendly interfaces for face enrollment, authentication, and management, I will enhance the user experience. Improve accessibility features so that people with a range of abilities and preferences can use FaceTrek Attendance.
- **Scalability and Deployment:** To manage workloads and resource needs, I will investigate cloud-based options for distributed FaceTrek Attendance processing.
- **Continuous Learning and Feedback:** In order to gather user input and performance information for continuous system improvement, I will put feedback systems in place. Allowing FaceTrek Attendance to adjust and improve its behavior in response to feedback and interactions from the real world by incorporating reinforcement learning techniques.

My FaceTrek Attendance - Face recognition Attendance System can be made more accurate, dependable, secure, and morally sound by concentrating on these areas of improvement, which will create new opportunities for the broad application.

Bibliography

References:

- https://stackoverflow.com/
- https://www.w3schools.com/
- https://opencv.org/
- https://docs.python.org/3/library/tkinter.html
- https://www.twilio.com/docs
- https://docs.python.org/3/library/csv.html
- https://gtts.readthedocs.io/en/latest/index.html
- https://www.mongodb.com/docs/
- https://docs.python.org/3/library/smtplib.html
- https://docs.python.org/3/library/shutil.html
- https://pythonbasics.org/python-play-sound/
- https://dutypar.com/blogs/how-does-face-recognition-attendance-system-work/
- https://www.jibble.io/article/face-recognition-attendance