Mini Project Report On

INGENIOUS APPROACH FOR ATTENDANCE USING LBPH ALGORITHM

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

Bachelor of Technology

In

Department of Computer Science and Engineering

By

A. SHASHIVADAN	**21245A0502**
A. SHIVAKUMAR	**21245A0503**
B. SUDHEER	**21245A0504**
D. ABHINAY	**21245A0506**

Under the Esteemed guidance of

G. NEELIMA

Assistant Professor

Department of Computer Science and Engineering

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

Mini Project Report On

INGENIOUS APPROACH FOR ATTENDANCE USING LBPH ALGORITHM

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

Bachelor of Technology

In

Department of Computer Science and Engineering

By

A. SHASHIVADAN	**21245A0502**
A. SHIVAKUMAR	**21245A0503**
B. SUDHEER	**21245A0504**
D. ABHINAY	**21245A0506**

Under the Esteemed guidance of

G. NEELIMA

Assistant Professor

Department of Computer Science and Engineering

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

GOKARAJU RANGARAJU

INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

CERTIFICATE

This is to certify that the mini project entitled "**INGENIOUS APPROACH FOR ATTENDANCE USING LBPH ALGORITHM**" is submitted by **A.SHASHIVADAN (21245A0502), A.SHIVAKUMAR (21245A0503), B.SUDHEER (21245A0504), D.ABHINAY (21245A0506)** in partial fulfilment of the award of degree in BACHELOR OF TECHNOLOGY in Computer Science and Engineering during academic year 2022-2023.

INTERNAL GUIDE

G. NEELIMA

Assistant Professor

HEAD OF THE DEPARTMENT

DR. K. MADHAVI

Professor & HOD

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

There are many people who helped us directly and indirectly to complete our project successfully. We would like to take this opportunity to thank one and all. First we wish to express our deep gratitude towards our internal guide **G. NEELIMA, Assistant Professor ,**Department of CSE for her support in the completion of our project report. We wish to express our honest and sincere thanks to **DR. K. Madhavi, HOD, Department of CSE** and to our principal **DR. J. Praveen** for providing the facilities to complete mini project. We would like to thank all our faculty and friends for their help and constructive criticism during the project completion phase. Finally, we are very much indebted to our parents for their moral support and encouragement to achieve goals.

A. SHASHIVADAN (21245A0502)
A. SHIVAKUMAR (21245A0503)
B. SUDHEER (21245A0504)
D. ABHINAY (21245A0506)

DECLARATION

We hereby declare that the mini project entitled **“INGENIOUS APPROACH FOR ATTENDANCE USING LBPH ALGORITHM”** is the work done during the period from **12th Jan 2023 to 3rd June 2023** and is submitted in the partial fulfilment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering from Gokaraju Rangaraju Institute of Engineering and Technology (Autonomous under Jawaharlal Nehru Technology University, Hyderabad).The results embodied in this project have not been submitted to any other university or Institution for the award of any degree or diploma.

A. SHASHIVADAN (21245A502)
A. SHIVAKUMAR (21245A0503)
B. SUDHEER (21245A0504)
D. ABHINAY (21245A0506)

Table of Contents

List of Figures:

List of Tables:

ABSTRACT

The Face Recognition Attendance System is a technology-based solution that uses machine learning algorithms to accurately identify individuals and record their attendance. This system utilizes facial recognition software to scan and match faces with pre-existing data to confirm the identity of the attendee. This process eliminates the need for traditional methods such as manual sign-ins and signature verification, resulting in a more efficient and secure system.

Convolutional Neural Networks (CNNs) are a type of deep learning neural network architecture. The key feature of a CNN is the use of convolutional layers, where each neuron in a layer is connected to a small region of the previous layer. This allows the network to learn a hierarchy of increasingly complex features, starting from simple edge detection to more abstract shapes.

The system can be integrated with various platforms, such as time and attendance software, to provide real-time updates on attendance records. With its ability to accurately identify individuals and automate the attendance process.

CHAPTER 1

INTRODUCTION

1.1 Existing System

In the existing system the face recognition attendance system is based on deep learning convolutional neural networks. The face is most used biometrics for human identity authentication. They used three pre-trained convolutional neural networks and trained their data. The three networks showed very high performance in terms of prediction accuracy and reasonable training time. The main advantage of existing system over manual attendance system is data security and cannot be manipulated.

1.1.1 Limitations to Existing System

Some of the limitations include:

- The memory consumption is huge.
- Computational complexity is more.
- Limited model capacity.
- Scalability is limited.

1.2 Proposed System

The proposed system involves LBPH and Haar cascade algorithm based system that is more efficient and scalable. The system is capable of verifying the person and providing the attendance if the data is matched. The data is trained using LBPH algorithm which is more efficient in terms of time and space complexity. The person just need to stand in front of camera and if the data is matched with the existing data then the attendance is updated in excel sheet. This system also provides student management where the admin can manage the student details like update, delete etc. This combination of student management and face recognition makes the system more flexible for the users.

1.2.1 Advantages over Existing System

Some of the advantages of the system are:

- It is more secure.
- It is more efficient.
- Memory consumption is less.
- Student management is also provided.

1.3 Introduction to Face Recognition Attendance System

The face recognition attendance system based on Local Binary Pattern Histogram (LBPH) algorithm and the Haar Cascade Frontal Face algorithm is efficient solution for automated attendance tracking. This system shows the power of computer vision techniques to accurately recognize and identify individuals based on their facial features. Because of the both LBPH and Haar Cascade Frontal Face algorithms the system achieves robust face recognition, making it very powerful and efficient attendance management solution.

The LBPH algorithm is widely used technique for facial recognition. It analyses the pixel intensity variations and generates a representation of an individual's facial features. This algorithm is very good at handling challenging factors such as changes in lighting conditions, facial expressions. This algorithm achieves high level of accuracy in matching extracted features with stored templates. This algorithm gives you reliable and consistent results, it reduces the false and error matches.

Haar Cascade Frontal Face algorithm plays a crucial role in detecting faces within images. It uses machine learning techniques to identify facial patterns. These algorithms find or extract the face features even in difficult situations like partially occluded etc. By identifying the facial regions the Haar Cascade Frontal Face algorithm eases the recognition process. It increases the efficiency and accuracy.

The integration of both LBPH and Haar Cascade Frontal Face algorithms empowers the facial attendance system to overcome the manual attendance management and provide very efficient way for managing the whole attendance process. Manipulation of the data becomes very less and minimizes the chances of false attendance. It provides security of the individual data.

There are several reasons why face recognition attendance system is needed.

- Accuracy and reliability: Face recognition technology offers a high level of accuracy in attendance management. The traditional methods are highly prone to error and they can be easily manipulated. The false rate is also high in traditional methods.

- Contactless solution: These days the hygiene is very important for everyone and many people prefer the work to be done without any physical contact. This face recognition attendance system provides contactless solution.

- Time and cost efficiency: In the manual or traditional attendance system it takes more time and more effort by the management. This system eliminates the unwanted effort and takes very less time to manage the attendance process.

- Enhanced security: As we know traditional attendance system can be manipulated by anyone by putting false attendance. But this system provides high security because every individual has unique facial features and no one can mark false attendance on behalf of someone else.

- Scalability and adaptability: This system is highly scalable to different organizational sizes and can adapt very well. Large number of users can be handled by this system. Many organizations like colleges, offices, businesses can use this system.

CHAPTER 2

LITERATURE STUDY

S. No.	Title of the paper	Authors	Journal Details
1	Parallel Faces Recognition Attendance System with Anti-Spoofing Using Convolutional Neural Network	Stephen Bassi Joseph , Emmanuel Gbenga Dada , Sanjay Misra , and Samuel Ajoka	Springer, 08 February 2022
2	Face Recognition Smart Attendance System using Deep Transfer Learning	KhawlaAlhanaeea ,MithaAlhammadia , Nahla Almenhalia , MaadShatnawi	Elsevier, 2021
3	Smart Attendance Monitoring System (SAMS): A Face Recognition based Attendance System for Classroom Environment	Shubhobrata Bhattacharya, Gowtham Sandeep Nainala, Prosenjit Das and AurobindaRoutray	IEEE, 2018
4	Study of Implementing Automated Attendance System Using Face Recognition Technique	Nirmalya Kar, Mrinal Kanti Debbarma, Ashim Saha, and Dwijen Rudra Pal	International Journal of Computer and Communication Engineering, Vol. 1, No. 2, July 2012
5	Implementation of classroom attendance system based on face recognition in class	Ajinkya Patil1 ,Mrudang Shukla2 1Mtech (E&TC), 2Assisstant Professor Symbiosis institute of Technology, Pune, Maharashtra, India	International Journal of Advances in Engineering & Technology, July, 2014

Table 2.1 Literature Survey

2.1 Parallel Faces Recognition Attendance System with Anti-Spoofing Using Convolutional Neural Network

AUTHORS: StephenBassi Joseph , Emmanuel Gbenga Dada , Sanjay Misra , and Samuel Ajoka

ABSTRACT: "In recent days computer vision is gaining a lot of attraction from large audience around the world. There are many applications of it like forensics, cyber security. To eliminate manual attendance system which is time consuming and not secure this paper presents face recognition attendance system which acts as conventional attendance system in organizations and classrooms. The main advantage of this system is it overcomes the difficulties which occur while detecting the facial features. This system accurately extracts the facial features and recognizes the identity of the person. This system showed high performance and worked very efficiently this shows the proposed system is a promising facial recognition technology**"**

2.2 Face Recognition Smart Attendance System using Deep Transfer Learning

AUTHORS: KhawlaAlhanaeea ,MithaAlhammadia , Nahla Almenhalia , MaadShatnawi

ABSTRACT: "From the past few years face identification has been considered as interesting and very accurate research domain as it plays a major role in authentication and recognition in attendance management and access control systems. The most followed approach is manual attendance system in many organization and educational institutions which is not very secure and time consuming. As the technology is evolved there are much identification techniques came to notice like RFID, eye tracking, voice recognition. Facial features recognition technique is used in this paper based on deep learning convolutional neural networks. The three networks used are AlexNet, GoogleNet, SqueezeNet which showed very high performance in terms of accuracy and training time."

2.3 Smart Attendance Monitoring System (SAMS): A Face Recognition based Attendance System for Classroom Environment

AUTHORS: Shubhobrata Bhattacharya, Gowtham Sandeep Nainala, Prosenjit Das and AurobindaRoutray

ABSTRACT: "In the present academic system, day to day attendance of students plays a vital role in performance assessment. The regular methods used or practiced in most of the educational institutions are by calling their specific names or roll numbers, which is more time taking process and prone to error or false attendance. This article presents the automatic attendance management system for speed and secured attendance of students. This system is

constructed by the integration of efficient components to make a portable device for managing the student's attendance using Face Recognition technology. To improve accuracy of the system, we used face tracking technique. . All we did was first detect the face using Viola & Jones idea as described and then, we used the correlation tracker from the dlib library to keep track of the face from frame to frame. In this approach we don't have to detect the face after transforming to a new frame this saves the system computational power."

2.4 Study of Implementing Automated Attendance System Using Face Recognition Technique

AUTHORS: Nirmalya Kar, Mrinal Kanti Debbarma, Ashim Saha, and Dwijen Rudra Pal

ABSTRACT: "The authentication is very important in any system control in computer based communication. In many recognitions techniques the face recognition is very important and widely used in many organizations, applications. This paper proposes a simple method for student's attendance using facial features recognition techniques. Student's Attendance System is integrated with the Personal Component Analysis (PCA) algorithm. In the classroom environment itself the system records the attendance of different individuals and stores the data in database. The information of the individuals is maintained like clock-in and clock-out time."

2.5 Implementation of classroom attendance system based on face recognition in class

AUTHORS: Ajinkya Patil1, Mrudang Shukla2 1Mtech (E&TC), 2Assisstant Professor Symbiosis institute of Technology, Pune, Maharashtra, India

ABSTRACT: "The face is the identity of a person. The methods to exploit this physical feature have seen a great change since the advent of image processing techniques. The attendance is taken in every schools, colleges and library. Traditional approach for attendance is professor calls student name & record attendance. It takes some time to record attendance. Suppose duration of class of one subject is about 50 minutes & to record attendance takes 5 to 10 minutes. For each lecture this is wastage of time. To avoid these losses, we are about use automatic process which is based on image processing. In this novel approach, we are using face detection & face recognition system. This face detection differentiates faces from non-faces and is therefore essential for accurate attendance. The other strategy involves face recognition for marking the student's attendance. The Raspberry pi module is used for face detection & recognition. The camera will be connected to the Raspberry pi module. The student database is collected. The database includes name of the students, there images & roll number. This raspberry pi module will be installed at the front side of class in such a way that we can capture entire class. Thus with the help of this system, time will be saved. With the help of this system, it is so convenient to record attendance. We can take attendance on any time."

CHAPTER 3

SYSTEM SPECIFICATIONS

3.1 Introduction

3.1.1 Purpose of requirements document:

The main purpose of requirements document is used to record the functional and non functional requirements for the software program that helps face recognition attendance system. It acts as a blueprint for team to develop the system and ensure that every person knows what needs to be done and how it functions.

1. **Clear communication:** The communication tool between stakeholders, developers, and designers is software requirements document. These documents define the objectives, functionalities and constraints of the system. By documenting the specific requirements related to the Haar Cascade Frontal Face and LBPH algorithms, it ensures that everyone have a good understanding of the system's functionalities and limitations.

2. **Functional and non-functional requirements:** The document describes both functional and non-functional requirements of the system. The special features or functionalities from the face recognition attendance system like face detection, recognition accuracy, attendance recording, and integration with existing system are described by functional requirements. The non-functional requirements describes such as performance, security, scalability, and usability. These requirements help us to improve the performance of the system and ensure that system reaches its required needs.

3. **System design and development:** The software requirements document also provides the foundation for system design and development. The document helps the developers on how to implement the Haar Cascade Frontal Face algorithm and Local binary pattern histogram (LBPH) algorithm. The technical requirements like hardware specifications, programming languages, platforms, and any other dependencies are also specified by the document so that team doesn't find any difficulties**.**

4. **Validation and testing:** The requirements document is also used for validation and testing activities. This helps testers to verify whether the system meets the desired requirements or need and ensures that both the algorithms are functioning correctly.

5. **Future enhancements and maintenance:** The document also serves as a reference for future enhancements, updates, and maintenance of the face recognition attendance system. This document becomes invaluable for long-term system management and evolution.

In summary, creating software requirements document is very efficient to establish clear communication and to reach the needs specified for the system. It enables successful implementation and management of the system.

3.1.2 Scope of the Product

The scope of the face recognition attendance system encompasses the functionalities, features, and boundaries of the product. This defines what system will do and what is aims to achieve.

1. **Attendance Tracking:** The responsibility of the system is to record the attendance of individuals by using facial recognition technology. It will capture and analyze facial features of the individuals.

2. **Face Detection:** The Haar Cascade Frontal Face algorithm is used for detecting the face within the images. This algorithm will extract the facial features of the individuals even in difficult situations.

3. **Face Recognition:** The system performs LBPH algorithm for face recognition. I will compare the extracted features of the individuals and identify them. The recognition process is very accurate and fast efficient.

4. **Attendance Management:** The system also provides the management system for administrator to manage the attendance details. The data is stored in database and the record of the attendance is stored in excel sheets.

5. **Integration:** The system is also capable to integrate with other systems and databases. The system can be used in many organizations.

3.1.3 Definitions, acronyms, and Abbreviations

1. **Face Recognition Attendance System:** A computer-based system that uses facial recognition technology to automate the process of attendance system. It captures the facial features and authenticates individuals.

2. **Haar Cascade Frontal Face Algorithm:** This is machine learning-based algorithm used for face detection. It extracts the facial features of the individuals and analyzes the images.

3. **LBPH:** LBPH stands for Local Binary Pattern Histogram. This technique used for face recognitions that captures the local texture information from images. It analyzes the pixel intensity variations to create unique representation of facial features.

4. **SQL:** Structured query language is open-source relational database management system that is used for storing and managing data.

3.1.4 Overview of the remainder of the document

The document for the face recognition attendance system provides a comprehensive overview of the system's design, functionality, and implementation. It serves as a guide for stakeholders, developers, and project teams involved in the development and deployment of the system.

3.2 General Description

3.2.1 Product perspective

The product perspective for a face recognition attendance system involves considering its role, functionality, and interactions within the organizational context. The main focus is on the system meets the needs of users, integrates with existing processes.

1. **Purpose and value:** This system is designed to automate the process of manual attendance system using face recognition technique. The primary goal of the system is accurately and efficiently records the attendance.

2. **User Interaction:** The user interaction is more important in this system. The individuals can mark their attendance by presenting their face to the system which consists of web cam. The user interface is user-friendly and easy to understand.

3. **Integration with existing systems:** This system can be integrated with many other systems in organizations. It is highly scalable and it may also connect with human resources (HR) systems to store attendance data.

4. **Scalability and Flexibility:** The system is designed to scale and adapt to the organization's requirements or needs. It can handle rush and maintain the efficiency.

5. **Data security and privacy:** It employs measures such as encryption, access controls, and secure data storage to protect sensitive attendance data. This makes the manipulation of the data very difficult so the data is secured.

6. **Reporting and analysis:** The system provides good reporting and analysis capabilities, allowing admin to generate attendance reports, track trends, and identifies patterns.

3.2.2 Product functions

1. **Face Detection:** For capturing images and detecting the faces Haar Cascade Frontal Face algorithm is used by the system. This function locates the presence of facial features in the input data.

2. **Face Recognition:** After the faces are detected, the LBPH algorithm is applied to analyze and recognize the facial features of the individual persons and it extracts the unique texture or patterns to recognize easily. By matching the features, the system can accurately identify the individuals.

3. **Attendance Recording:** The system records the attendance by recognizing the faces of individuals and stores the information of the person in database or excel sheet. This function ensures real-time and accurate tracking of attendance.

4. **Exception Handling:** Exceptions are also handled where face detection or recognition may fail or produce inaccurate results. This helps the users understand the problem and solve the problem according to the system.

5. **Data Management:** The system manages the attendance data like storing, updating, deleting, organizing. This function ensures efficient data management, retrieval and tracking of attendance records.

3.2.3 User Characteristics

The face recognition attendance system can be used by different user roles, each with different requirements.

1. **Employees:** The primary uses of the face recognition attendance system are employees. They have various characteristics like time efficiency, privacy concerns.

2. **Administrators:** The responsibility of the administrators is managing the face recognition attendance system.

3. **It Support Staff:** IT support staff should be capable of handling technical challenges, performing system updates and addressing.

4. **System Administrators/Developers:** System administrators or developers are responsible for configuration, customization and maintenance of the face recognition attendance system.

5. **Technical Expertise:** Every developer should have very good understanding of architecture, algorithms, and latest technologies. They must ensure the system's optimal performance.

6. **Technical Competence:** Administrators need to be familiar with system's functionalities and features. They should be able to navigate the system, generate reports, manage user profiles, and handle exceptions.

3.2.4 Assumptions and Dependencies

3.2.4.1 Assumptions

When face recognition attendance system is to be developed some assumptions are made to guide the design and implementation process. These assumptions provide a basis for the system's functionality and behaviour.

1. **Adequate Image Quality:** The system assumes that the captured images will have good quality for face detection and recognition. Clear images help extract accurate facial features.

2. **Cooperative User Behaviour:** The system assumes that users, such as employees, will cooperate and position their faces appropriately for the system to capture and analyze facial data.

3. **Unique Facial Features:** The system assumes that each individual has different features that can be effectively captured and distinguished by the face recognition algorithm.

4. **Consistency of Facial Appearance:** The system assumes that an individual's face expressions remain consistent over time. The minor changes will not affect the system performance.

5. **Proper Database Management:** The attendance records must be properly managed and secured. This management is very crucial in order to maintain the database efficient and retrieval must be easy.

6. **Compliance with Privacy Regulations:** The system assumes compliance with relevant privacy regulations and policies. It assumes that appropriate consent has been obtained from individuals whose facial data is being captured and processed.

3.2.4.2 Dependencies

1. **Camera and Image Device:** The System depends on the camera for capturing the images for face detection and recognition.

2. **Computing Infrastructure:** The system requires an appropriate computing infrastructure to process and analyse the captured facial data.

3. **Face Recognition Algorithm:** The system depends on such as Haar Cascade Frontal Face algorithm and LBPH algorithm, for face detection and recognition.

4. **Operating System:** The system relies on specific operating system that supports required software components and interfaces with the hardware devices.

5. **Database Management System:** The system requires a database management system to store and retrieve attendance records, user profiles, and other relevant data.

3.3 Specific Requirements

3.3.1 Functional Requirements

- **Enrollment:** The system should allow administrators to enrol employees or students by capturing their facial images. It should generate and store unique facial templates for each enrolled employee.
- **Attendance Marking:** The system should enable individuals to mark their attendance by presenting their face to the camera. It should accurately detect and recognize the faces of the individuals.
- **Attendance Tracking:** The system should maintain attendance log that includes every details like dates, time, and individual details for each attendance marking.
- **Reporting and Analysis:** The system should generate monthly summary reports of the attendance records. It should be easy for retrieval and easy to access.
- **User Management:** The system should allow administrators to manage the data including operations like updating, deleting, removing profiles. It should provide options for assigning user roles and permissions to control system.
- **System Configuration and Settings:** The system should allow administrators to customize system preferences and settings based on organizational requirements.

3.3.2 Non Functional Requirements

- **Performance:** The system should take very less time for detection of face and recognizing the face of individual. The system should be capable of handling a large number of concurrent users and processing a high volume of face images without any degradation of the system performance.

- **Scalability:** The system should scale effectively to accommodate an increasing number of users and attendance data while maintaining optimized performance.
- **Security:** The system should ensure the privacy and protection of facial data and attendance records from manipulation of the data. The system should only be maintained or accessed by administrators.
- **Usability:** The system should have a user-friendly interface for easy accessing of the data and the system. Managing of the data must be easy and users should not face any difficulties while using the system.
- **Reliability and Availability:** The system should have a high availability rate, minimizing downtime and ensuring operation for attendance marking and data management. The system should minimize failures and fault attendance.
- **Maintainability:** The system should be designed with modular components, allowing for easy maintenance, updates, and future enhancements.

These non-functional requirements help define the performance, security, usability, reliability, and maintainability expectations of the face recognition attendance system. The system should operate effectively, securely and with optimal user experience.

CHAPTER 4

DESIGN

4.1 Project Description

4.1.1 System Data-flow Diagram

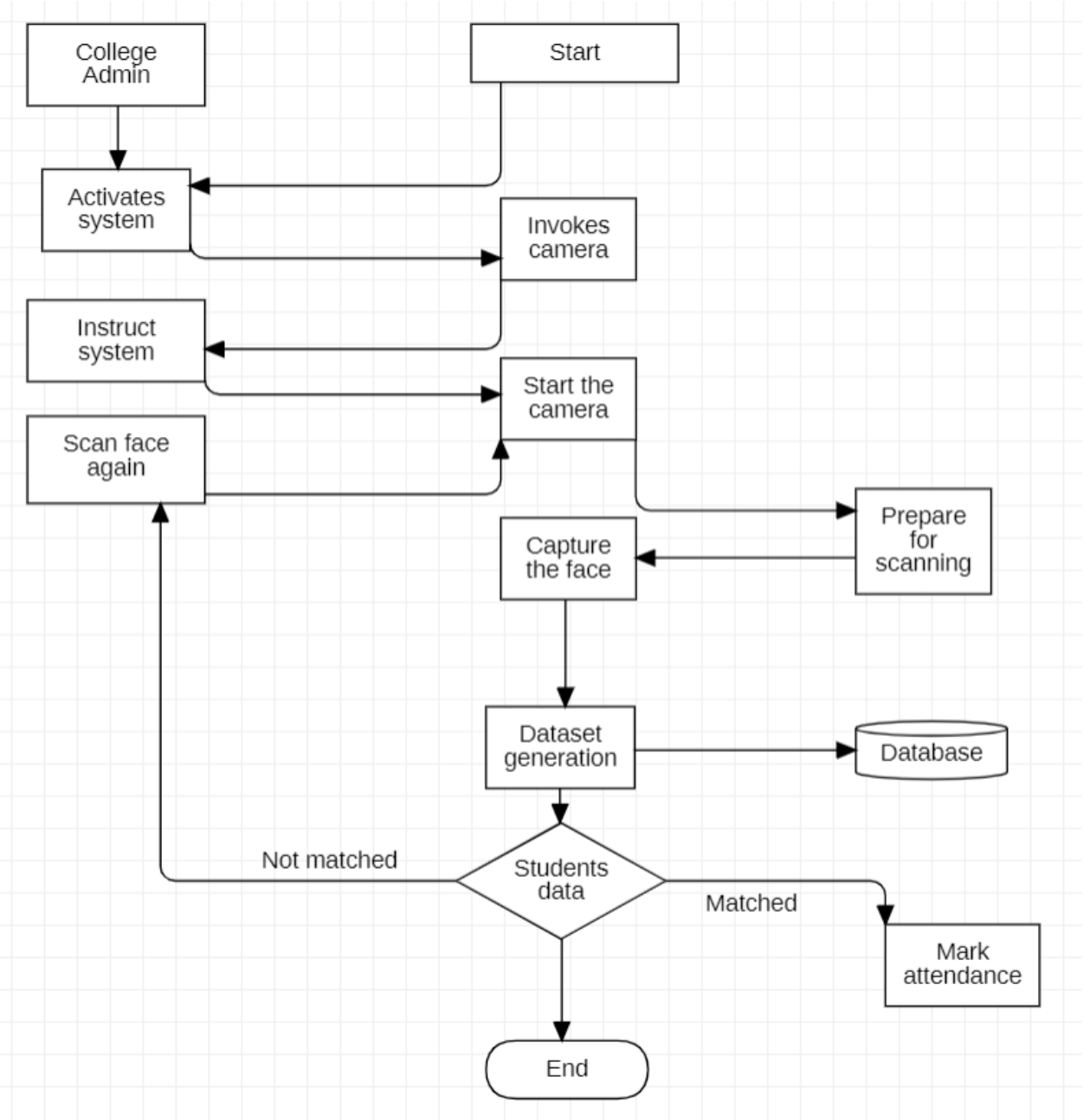


Fig 4.1. Dataflow Diagram

In the above data flow diagram, we have an admin to access the system. For detection and recognition the individual must present their face in front of the camera so that the camera takes the image of the face and analyzes it. If the image data is match with the existing data then the attendance is provided and updated in the database or excel sheet.

4.1.2 Use case Diagram

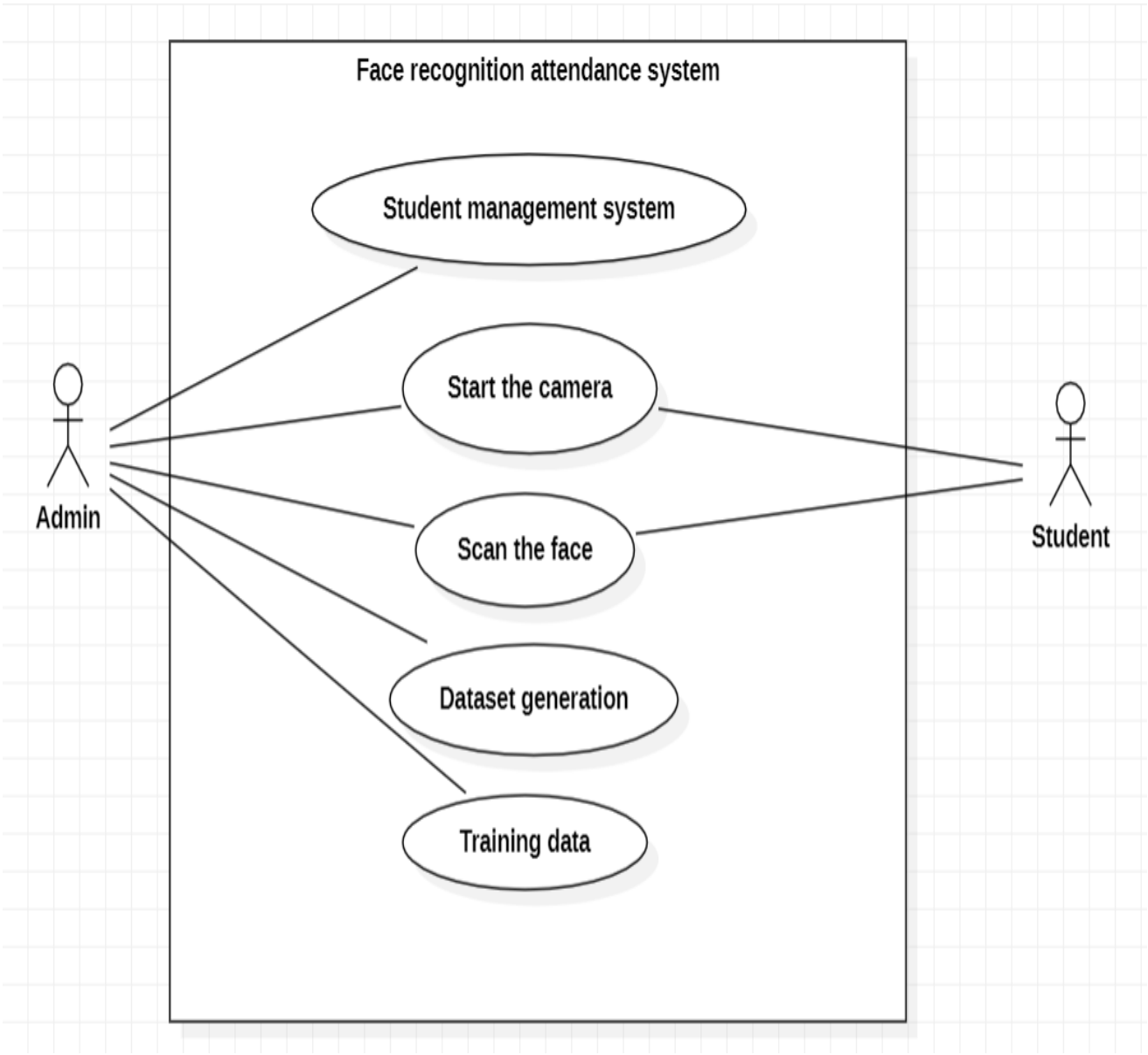


Fig 4.2. Class Diagram

The use case diagram gives the various operations in the system. It shows the various attributes and operations the system contains. It also shows the relation between multiple actors and functions. The use case diagram simplifies the construction of the system.

4.1.3 Sequence Diagram

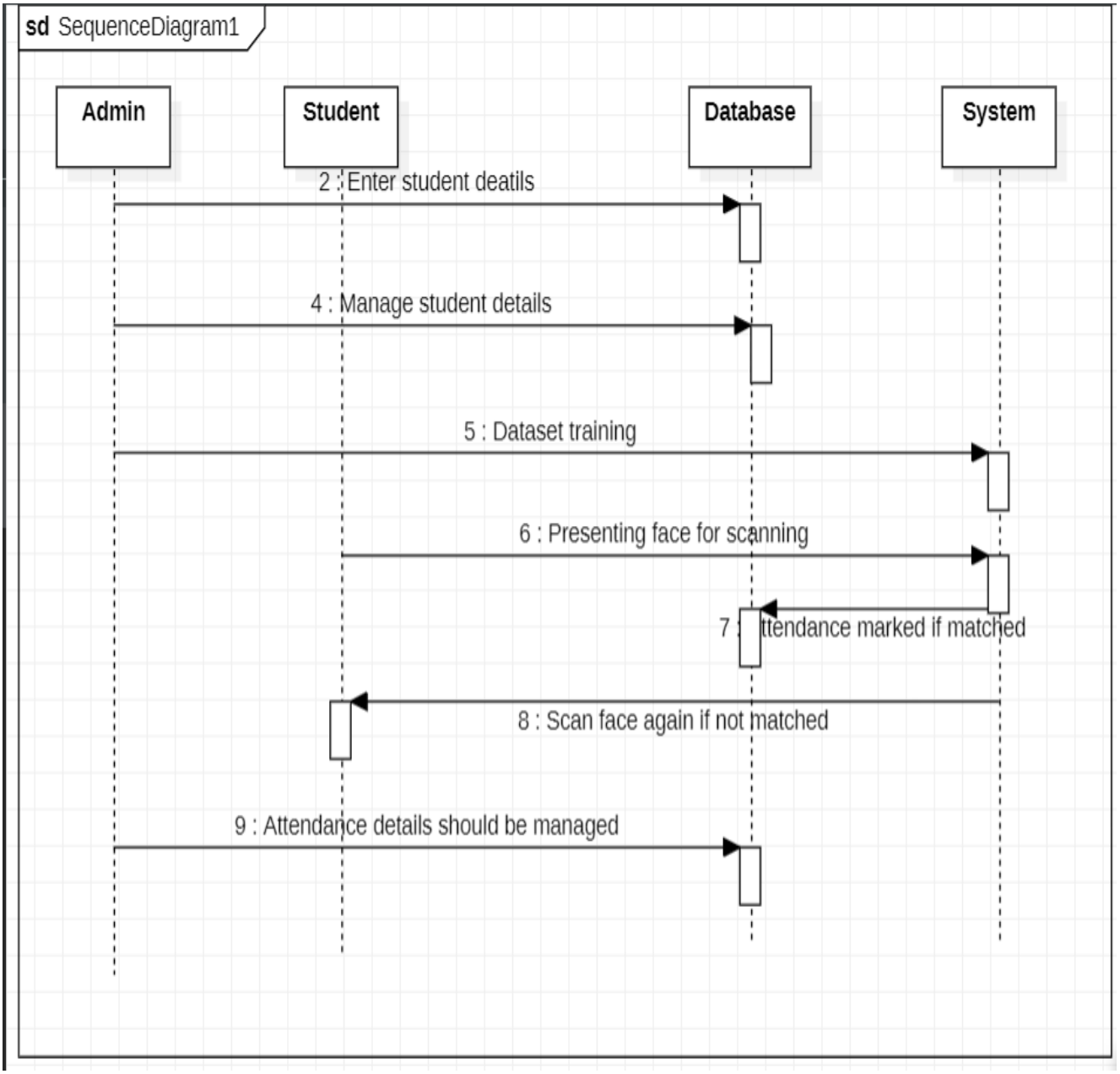


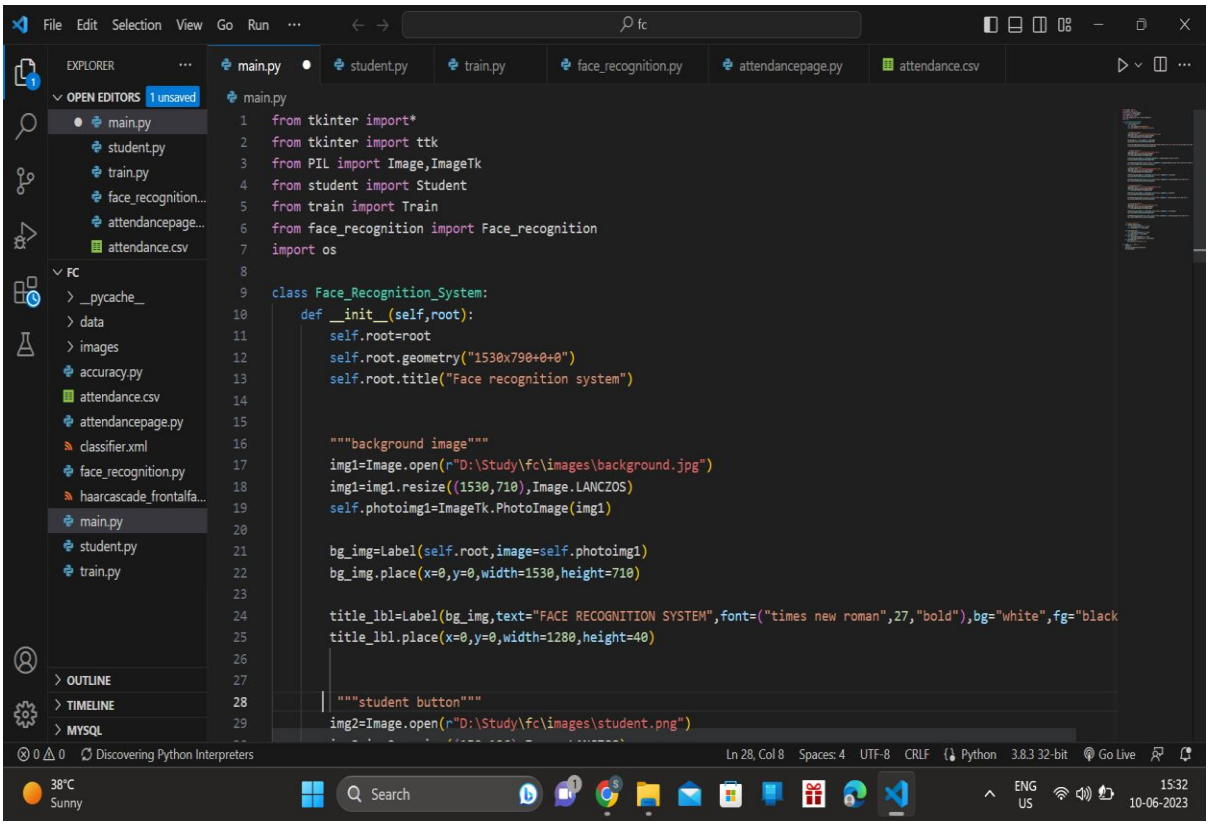
Fig 4.3. Sequence Diagram

The sequence diagram shows how the flow goes from one operation to another. In the sequence diagram it contains admin, student, database, System as attributes. The sequence diagrams makes the work easy and understandable for the developers.

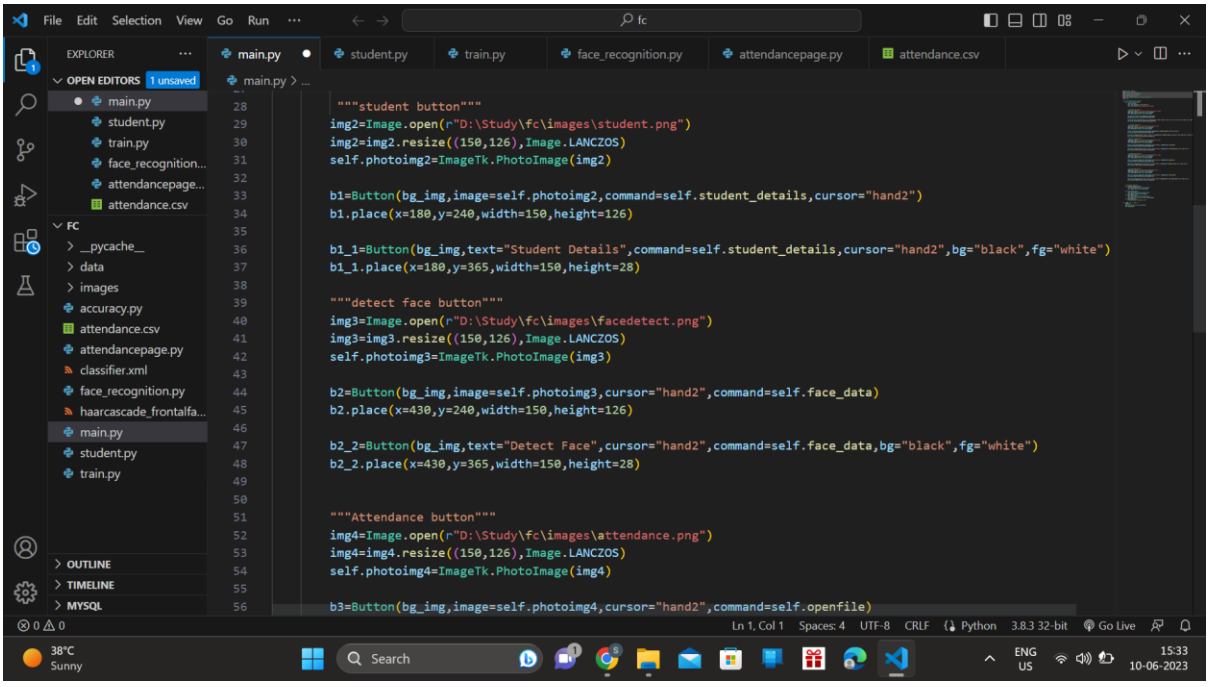
CHAPTER 5

IMPLEMENTATION

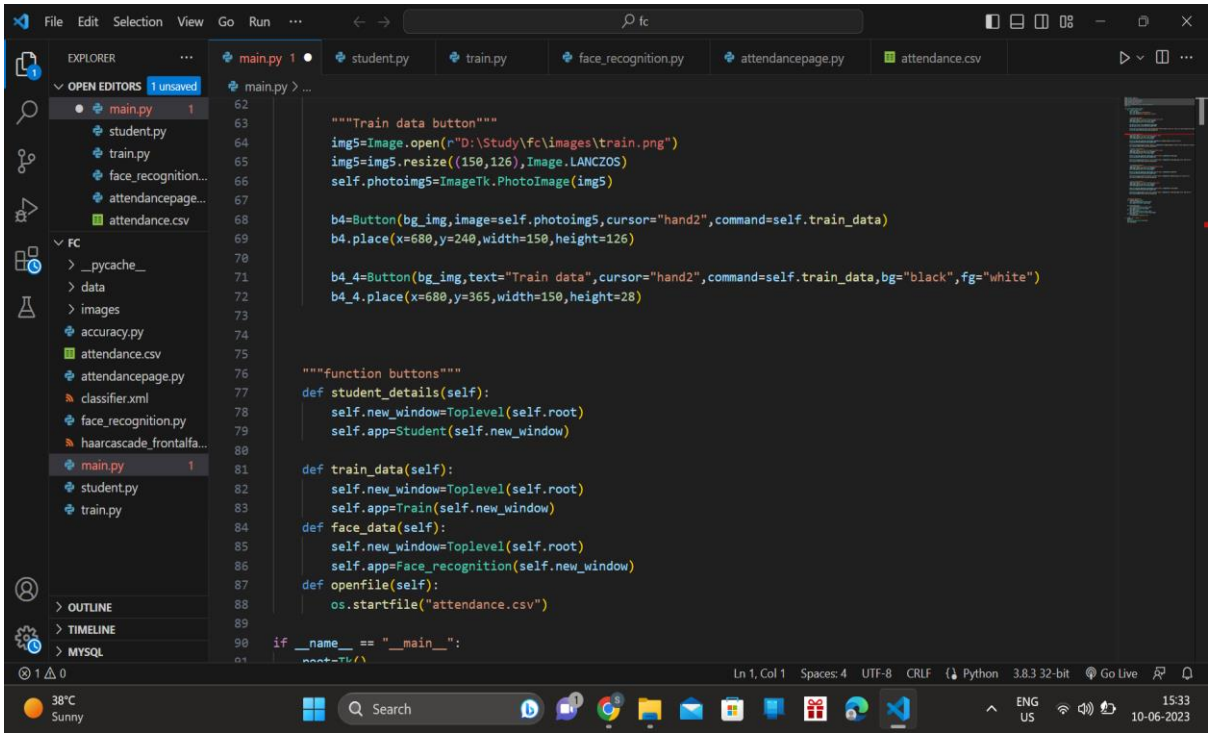
5.1 Source code



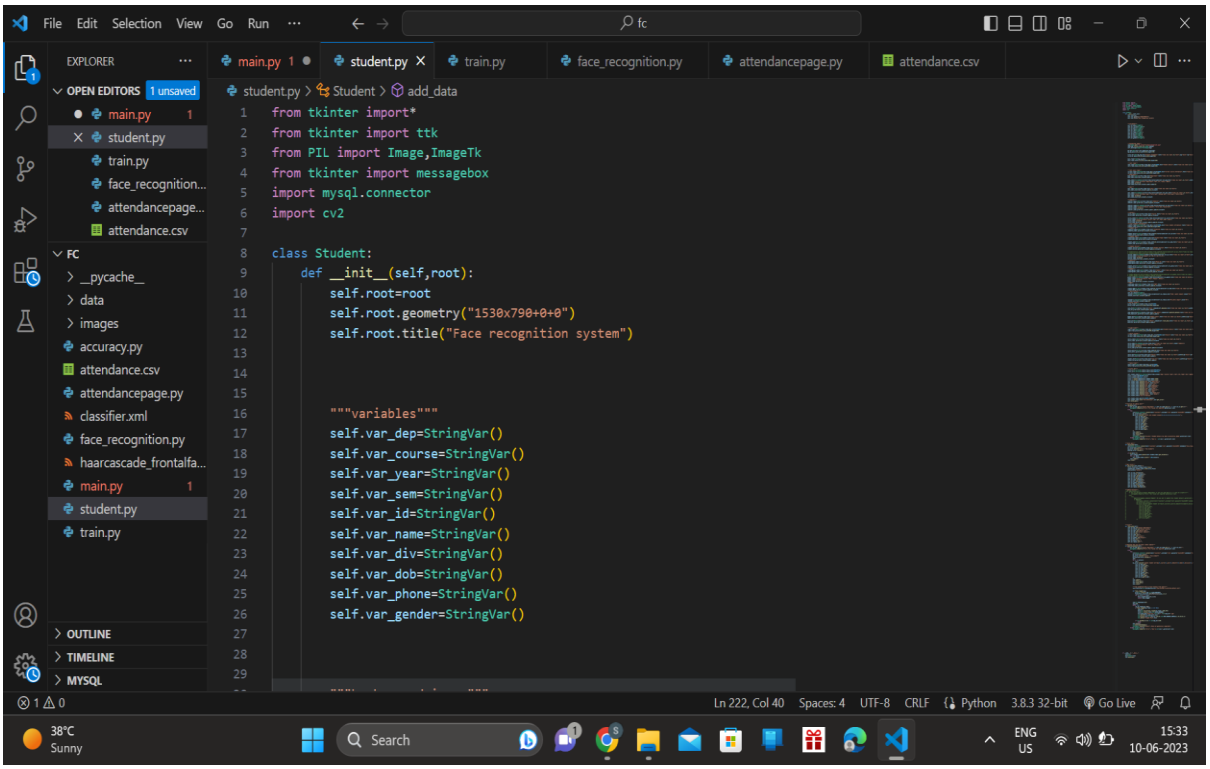
```
from tkinter import*
from tkinter import ttk
from PIL import Image,ImageTk
from student import Student
from train import Train
from face_recognition import Face_recognition
import os

class Face_Recognition_System:
    def __init__(self,root):
        self.root=root
        self.root.geometry("1530x790+0+0")
        self.root.title("Face recognition system")


        """background image"""
        img1=Image.open(r"D:\Study\fc\images\background.jpg")
        img1=img1.resize((1530,710),Image.LANCZOS)
        self.photoimg1=ImageTk.PhotoImage(img1)

        bg_img=Label(self.root,image=self.photoimg1)
        bg_img.place(x=0,y=0,width=1530,height=710)

        title_lbl=Label(bg_img,text="FACE RECOGNITION SYSTEM",font=("times new roman",27,"bold"),bg="white",fg="black
        title_lbl.place(x=0,y=0,width=1280,height=40)


        """student button"""
        img2=Image.open(r"D:\Study\fc\images\student.png")
```



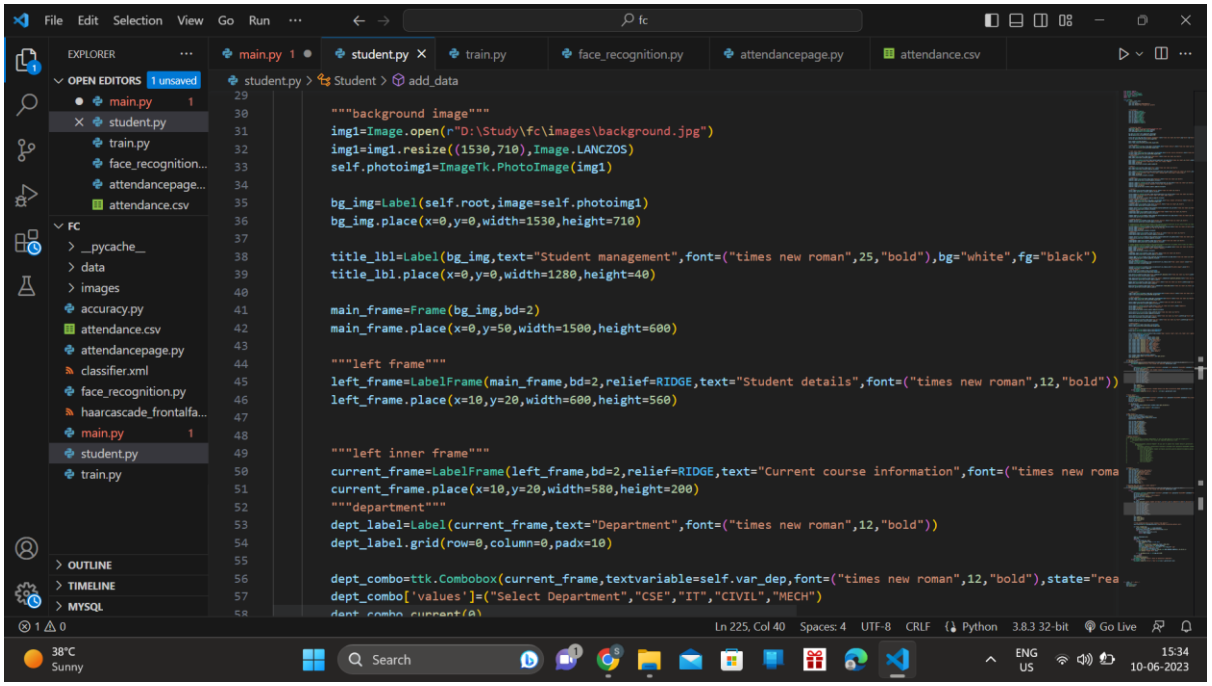
```
        """student button"""
        img2=Image.open(r"D:\Study\fc\images\student.png")
        img2=img2.resize((150,126),Image.LANCZOS)
        self.photoimg2=ImageTk.PhotoImage(img2)

        b1=Button(bg_img,image=self.photoimg2,command=self.student_details,cursor="hand2")
        b1.place(x=180,y=240,width=150,height=126)

        b1_1=Button(bg_img,text="Student Details",command=self.student_details,cursor="hand2",bg="black",fg="white")
        b1_1.place(x=180,y=365,width=150,height=28)

        """detect face button"""
        img3=Image.open(r"D:\Study\fc\images\facedetect.png")
        img3=img3.resize((150,126),Image.LANCZOS)
        self.photoimg3=ImageTk.PhotoImage(img3)

        b2=Button(bg_img,image=self.photoimg3,cursor="hand2",command=self.face_data)
        b2.place(x=430,y=240,width=150,height=126)

        b2_2=Button(bg_img,text="Detect Face",cursor="hand2",command=self.face_data,bg="black",fg="white")
        b2_2.place(x=430,y=365,width=150,height=28)


        """Attendance button"""
        img4=Image.open(r"D:\Study\fc\images\attendance.png")
        img4=img4.resize((150,126),Image.LANCZOS)
        self.photoimg4=ImageTk.PhotoImage(img4)

        b3=Button(bg_img,image=self.photoimg4,cursor="hand2",command=self.openfile)
```



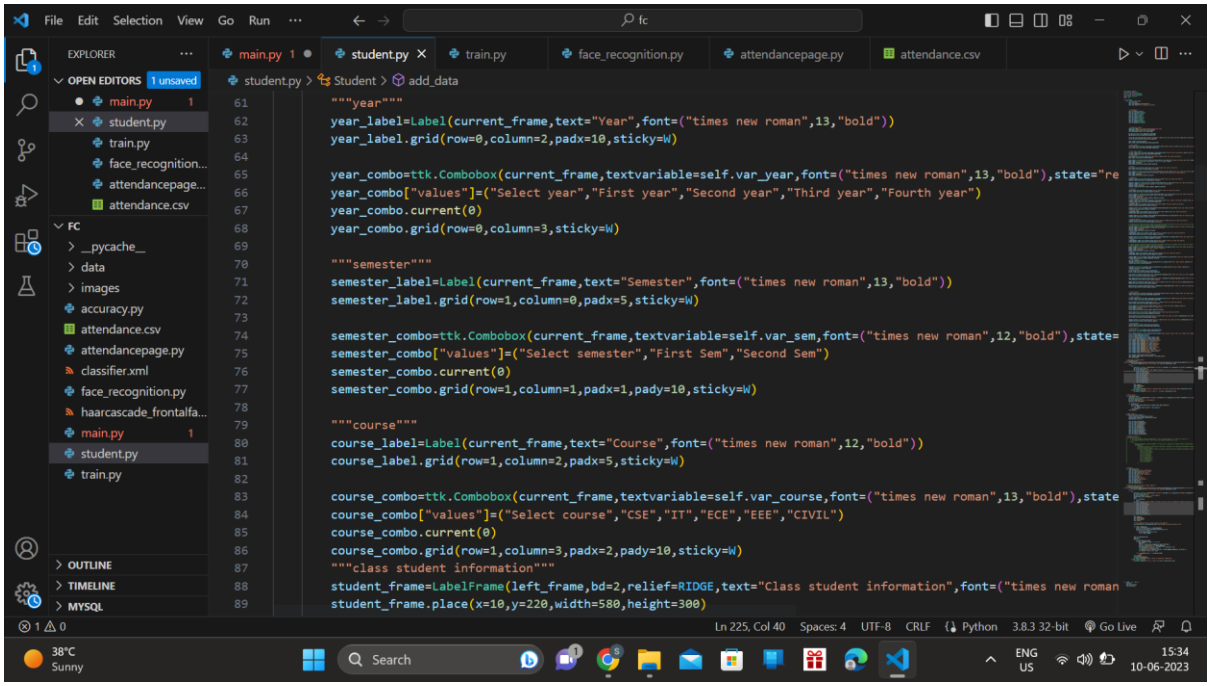
```
        """Train data button"""
        img5=Image.open(r"D:\Study\fc\images\train.png")
        img5=img5.resize((150,126),Image.LANCZOS)
        self.photoimg5=ImageTk.PhotoImage(img5)

        b4=Button(bg_img,image=self.photoimg5,cursor="hand2",command=self.train_data)
        b4.place(x=680,y=240,width=150,height=126)

        b4_4=Button(bg_img,text="Train data",cursor="hand2",command=self.train_data,bg="black",fg="white")
        b4_4.place(x=680,y=365,width=150,height=28)



    """function buttons"""
    def student_details(self):
        self.new_window=Toplevel(self.root)
        self.app=Student(self.new_window)

    def train_data(self):
        self.new_window=Toplevel(self.root)
        self.app=Train(self.new_window)
    def face_data(self):
        self.new_window=Toplevel(self.root)
        self.app=Face_recognition(self.new_window)
    def openfile(self):
        os.startfile("attendance.csv")

if __name__ == "__main__":
```



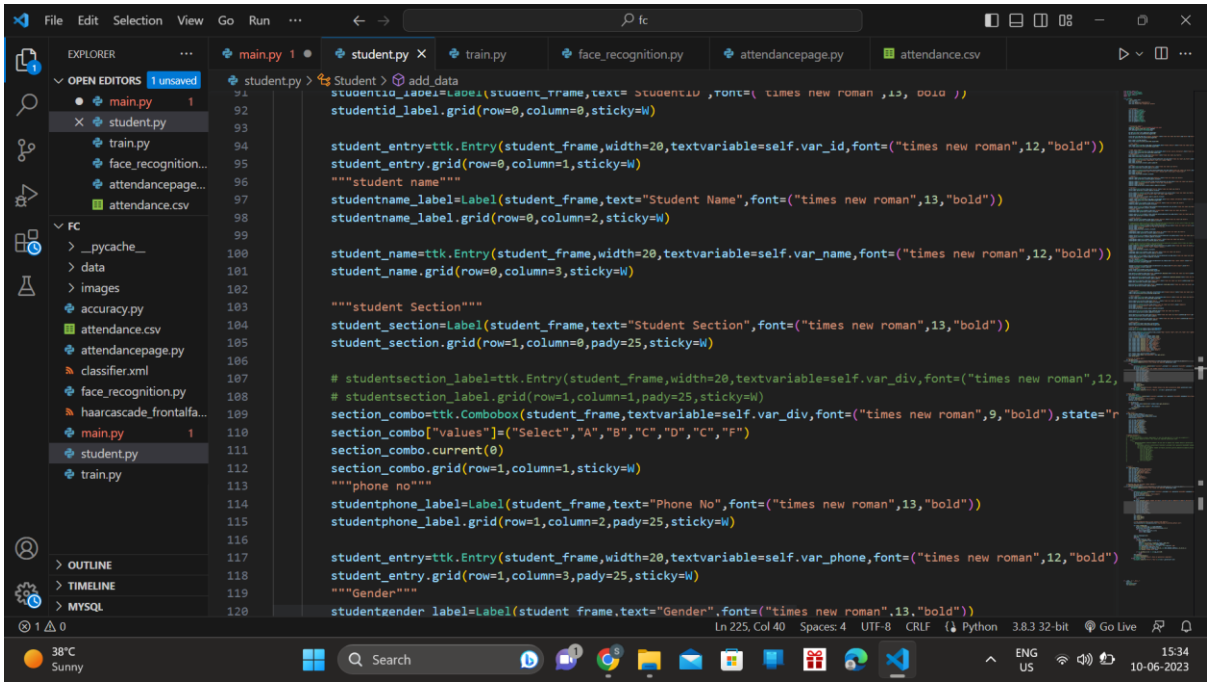
```
from tkinter import*
from tkinter import ttk
from PIL import Image,ImageTk
from tkinter import messagebox
import mysql.connector
import cv2

class Student:
    def __init__(self,root):
        self.root=root
        self.root.geometry("1530x790+0+0")
        self.root.title("Face recognition system")



        """variables"""
        self.var_dep=StringVar()
        self.var_course=StringVar()
        self.var_year=StringVar()
        self.var_sem=StringVar()
        self.var_id=StringVar()
        self.var_name=StringVar()
        self.var_div=StringVar()
        self.var_dob=StringVar()
        self.var_phone=StringVar()
        self.var_gender=StringVar()
```



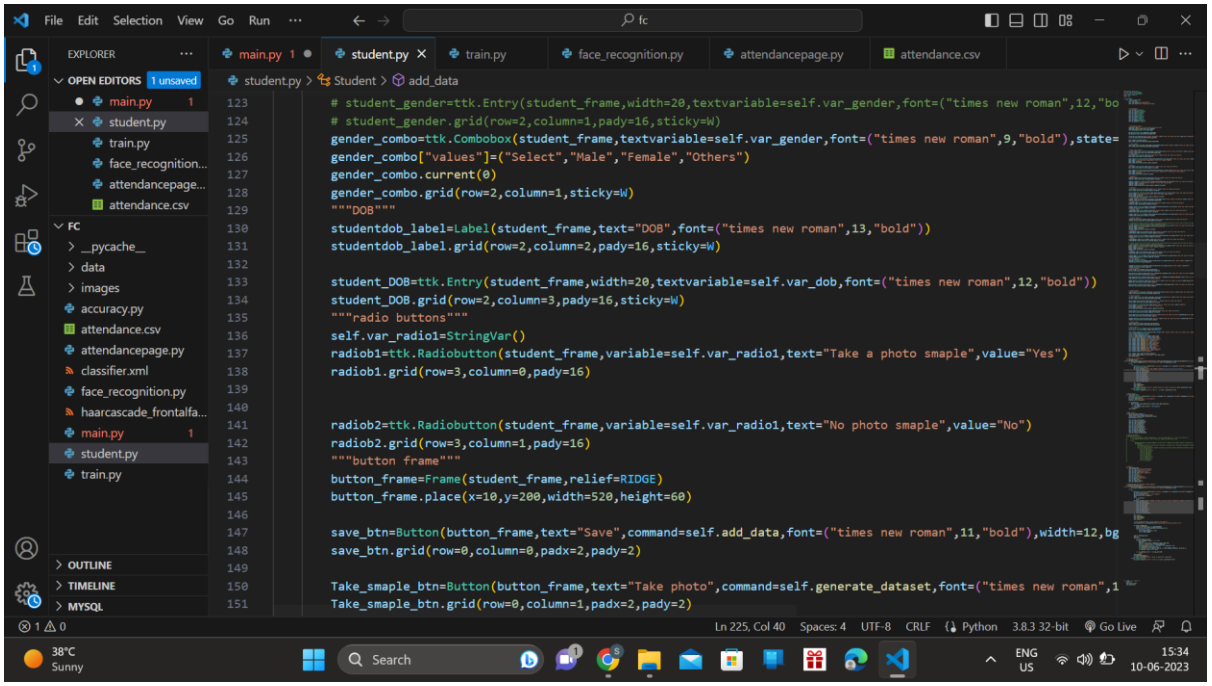
```
        """background image"""
        img1=Image.open(r"D:\Study\fc\images\background.jpg")
        img1=img1.resize((1530,710),Image.LANCZOS)
        self.photoimg1=ImageTk.PhotoImage(img1)

        bg_img=Label(self.root,image=self.photoimg1)
        bg_img.place(x=0,y=0,width=1530,height=710)

        title_lbl=Label(bg_img,text="Student management",font=("times new roman",25,"bold"),bg="white",fg="black")
        title_lbl.place(x=0,y=0,width=1280,height=40)

        main_frame=Frame(bg_img,bd=2)
        main_frame.place(x=0,y=50,width=1500,height=600)

        """left frame"""
        left_frame=LabelFrame(main_frame,bd=2,relief=RIDGE,text="Student details",font=("times new roman",12,"bold"))
        left_frame.place(x=10,y=20,width=600,height=560)


        """left inner frame"""
        current_frame=LabelFrame(left_frame,bd=2,relief=RIDGE,text="Current course information",font=("times new roma
        current_frame.place(x=10,y=20,width=580,height=200)
        """department"""
        dept_label=Label(current_frame,text="Department",font=("times new roman",12,"bold"))
        dept_label.grid(row=0,column=0,padx=10)

        dept_combo=ttk.Combobox(current_frame,textvariable=self.var_dep,font=("times new roman",12,"bold"),state="rea
        dept_combo['values']=("Select Department","CSE","IT","CIVIL","MECH")
```



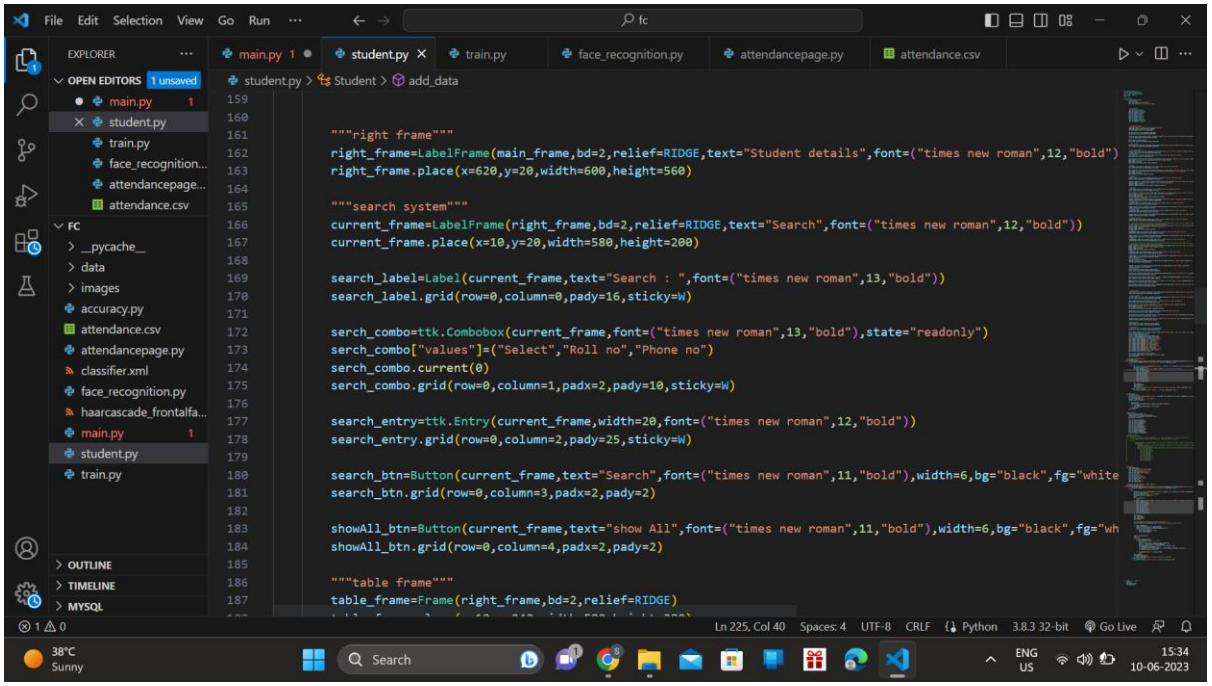
```
        """year"""
        year_label=Label(current_frame,text="Year",font=("times new roman",13,"bold"))
        year_label.grid(row=0,column=2,padx=10,sticky=W)

        year_combo=ttk.Combobox(current_frame,textvariable=self.var_year,font=("times new roman",13,"bold"),state="re
        year_combo["values"]=("Select year","First year","Second year","Third year","Fourth year")
        year_combo.current(0)
        year_combo.grid(row=0,column=3,sticky=W)

        """semester"""
        semester_label=Label(current_frame,text="Semester",font=("times new roman",13,"bold"))
        semester_label.grid(row=1,column=0,padx=5,sticky=W)

        semester_combo=ttk.Combobox(current_frame,textvariable=self.var_sem,font=("times new roman",12,"bold"),state=
        semester_combo["values"]=("Select semester","First Sem","Second Sem")
        semester_combo.current(0)
        semester_combo.grid(row=1,column=1,padx=1,pady=10,sticky=W)

        """course"""
        course_label=Label(current_frame,text="Course",font=("times new roman",12,"bold"))
        course_label.grid(row=1,column=2,padx=5,sticky=W)

        course_combo=ttk.Combobox(current_frame,textvariable=self.var_course,font=("times new roman",13,"bold"),state
        course_combo["values"]=("Select course","CSE","IT","ECE","EEE","CIVIL")
        course_combo.current(0)
        course_combo.grid(row=1,column=3,padx=2,pady=10,sticky=W)
        """class student information"""
        student_frame=LabelFrame(left_frame,bd=2,relief=RIDGE,text="Class student information",font=("times new roman
        student_frame.place(x=10,y=220,width=580,height=300)
```

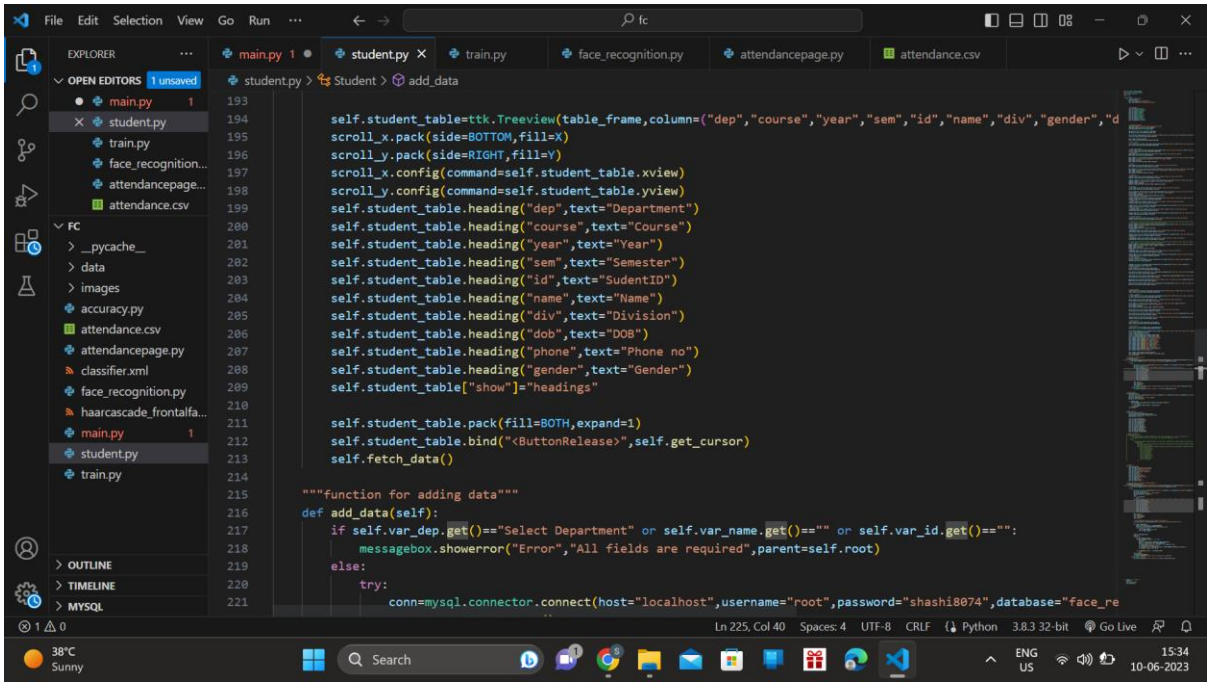


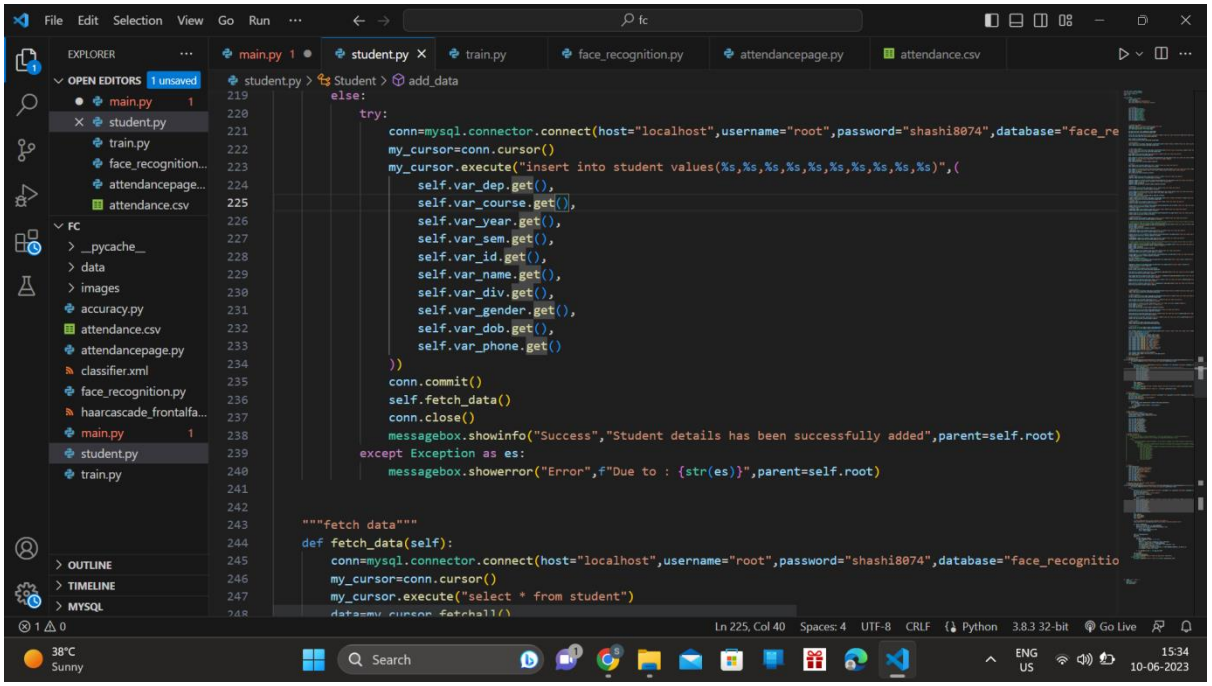
```
        studentid_label.grid(row=0,column=0,sticky=W)

        student_entry=ttk.Entry(student_frame,width=20,textvariable=self.var_id,font=("times new roman",12,"bold"))
        student_entry.grid(row=0,column=1,sticky=W)
        """student name"""
        studentname_label=Label(student_frame,text="Student Name",font=("times new roman",13,"bold"))
        studentname_label.grid(row=0,column=2,sticky=W)

        student_name=ttk.Entry(student_frame,width=20,textvariable=self.var_name,font=("times new roman",12,"bold"))
        student_name.grid(row=0,column=3,sticky=W)

        """student Section"""
        student_section=Label(student_frame,text="Student Section",font=("times new roman",13,"bold"))
        student_section.grid(row=1,column=0,pady=25,sticky=W)

        # studentsection_label=ttk.Entry(student_frame,width=20,textvariable=self.var_div,font=("times new roman",12,
        # studentsection_label.grid(row=1,column=1,pady=25,sticky=W)
        section_combo=ttk.Combobox(student_frame,textvariable=self.var_div,font=("times new roman",9,"bold"),state="r
        section_combo["values"]=("Select","A","B","C","D","C","F")
        section_combo.current(0)
        section_combo.grid(row=1,column=1,sticky=W)
        """phone no"""
        studentphone_label=Label(student_frame,text="Phone No",font=("times new roman",13,"bold"))
        studentphone_label.grid(row=1,column=2,pady=25,sticky=W)

        student_entry=ttk.Entry(student_frame,width=20,textvariable=self.var_phone,font=("times new roman",12,"bold")
        student_entry.grid(row=1,column=3,pady=25,sticky=W)
        """Gender"""
        studentgender_label=Label(student_frame,text="Gender",font=("times new roman",13,"bold"))
```

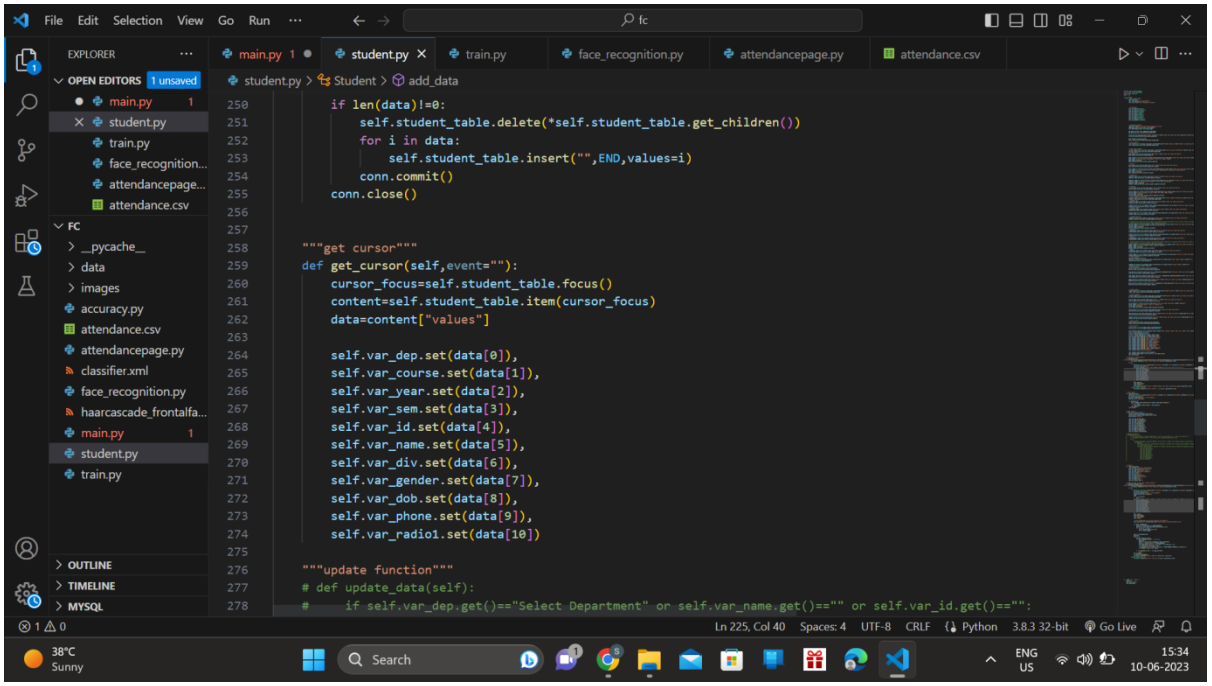


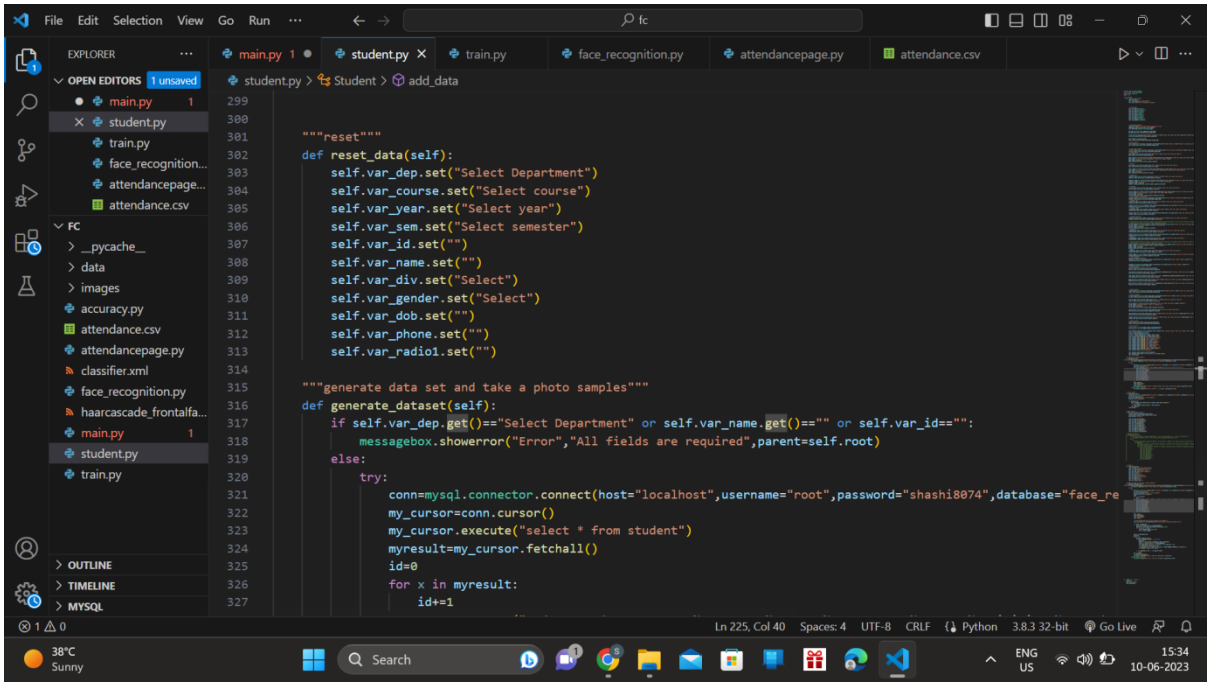
```
        # student_gender=ttk.Entry(student_frame,width=20,textvariable=self.var_gender,font=("times new roman",12,"bo
        # student_gender.grid(row=2,column=1,pady=16,sticky=W)
        gender_combo=ttk.Combobox(student_frame,textvariable=self.var_gender,font=("times new roman",9,"bold"),state=
        gender_combo["values"]=("Select","Male","Female","Others")
        gender_combo.current(0)
        gender_combo.grid(row=2,column=1,sticky=W)
        """DOB"""
        studentdob_label=Label(student_frame,text="DOB",font=("times new roman",13,"bold"))
        studentdob_label.grid(row=2,column=2,pady=16,sticky=W)

        student_DOB=ttk.Entry(student_frame,width=20,textvariable=self.var_dob,font=("times new roman",12,"bold"))
        student_DOB.grid(row=2,column=3,pady=16,sticky=W)
        """radio buttons"""
        self.var_radio1=StringVar()
        radiob1=ttk.Radiobutton(student_frame,variable=self.var_radio1,text="Take a photo smaple",value="Yes")
        radiob1.grid(row=3,column=0,pady=16)


        radiob2=ttk.Radiobutton(student_frame,variable=self.var_radio1,text="No photo smaple",value="No")
        radiob2.grid(row=3,column=1,pady=16)
        """button frame"""
        button_frame=Frame(student_frame,relief=RIDGE)
        button_frame.place(x=10,y=200,width=520,height=60)

        save_btn=Button(button_frame,text="Save",command=self.add_data,font=("times new roman",11,"bold"),width=12,bg
        save_btn.grid(row=0,column=0,padx=2,pady=2)

        Take_smaple_btn=Button(button_frame,text="Take photo",command=self.generate_dataset,font=("times new roman",1
        Take_smaple_btn.grid(row=0,column=1,padx=2,pady=2)
```


File Edit Selection View Go Run
EXPLORER
OPEN EDITORS 1 unsaved
main.py 1
student.py
train.py
face_recognition...
attendancepage...
attendance.csv
FC
__pycache__
data
images
accuracy.py
attendance.csv
attendancepage.py
classifier.xml
face_recognition.py
haarcascade_frontalfa...
main.py 1
student.py
train.py
OUTLINE
TIMELINE
MYSQL
main.py 1
student.py
train.py
face_recognition.py
attendancepage.py
attendance.csv
student.py > Student > add_data
"""right frame"""
right_frame=LabelFrame(main_frame,bd=2,relief=RIDGE,text="Student details",font=("times new roman",12,"bold")
right_frame.place(x=620,y=20,width=600,height=560)
"""search system"""
current_frame=LabelFrame(right_frame,bd=2,relief=RIDGE,text="Search",font=("times new roman",12,"bold"))
current_frame.place(x=10,y=20,width=580,height=200)
search_label=Label(current_frame,text="Search : ",font=("times new roman",13,"bold"))
search_label.grid(row=0,column=0,pady=16,sticky=W)
serch_combo=ttk.Combobox(current_frame,font=("times new roman",13,"bold"),state="readonly")
serch_combo["values"]=("Select","Roll no","Phone no")
serch_combo.current(0)
serch_combo.grid(row=0,column=1,padx=2,pady=10,sticky=W)
search_entry=ttk.Entry(current_frame,width=20,font=("times new roman",12,"bold"))
search_entry.grid(row=0,column=2,pady=25,sticky=W)
search_btn=Button(current_frame,text="Search",font=("times new roman",11,"bold"),width=6,bg="black",fg="white
search_btn.grid(row=0,column=3,padx=2,pady=2)
showAll_btn=Button(current_frame,text="show All",font=("times new roman",11,"bold"),width=6,bg="black",fg="wh
showAll_btn.grid(row=0,column=4,padx=2,pady=2)
"""table frame"""
table_frame=Frame(right_frame,bd=2,relief=RIDGE)
Ln 225, Col 40 Spaces: 4 UTF-8 CRLF Python 3.8.3 32-bit Go Live
38°C Sunny
Search
ENG US
15:34 10-06-2023

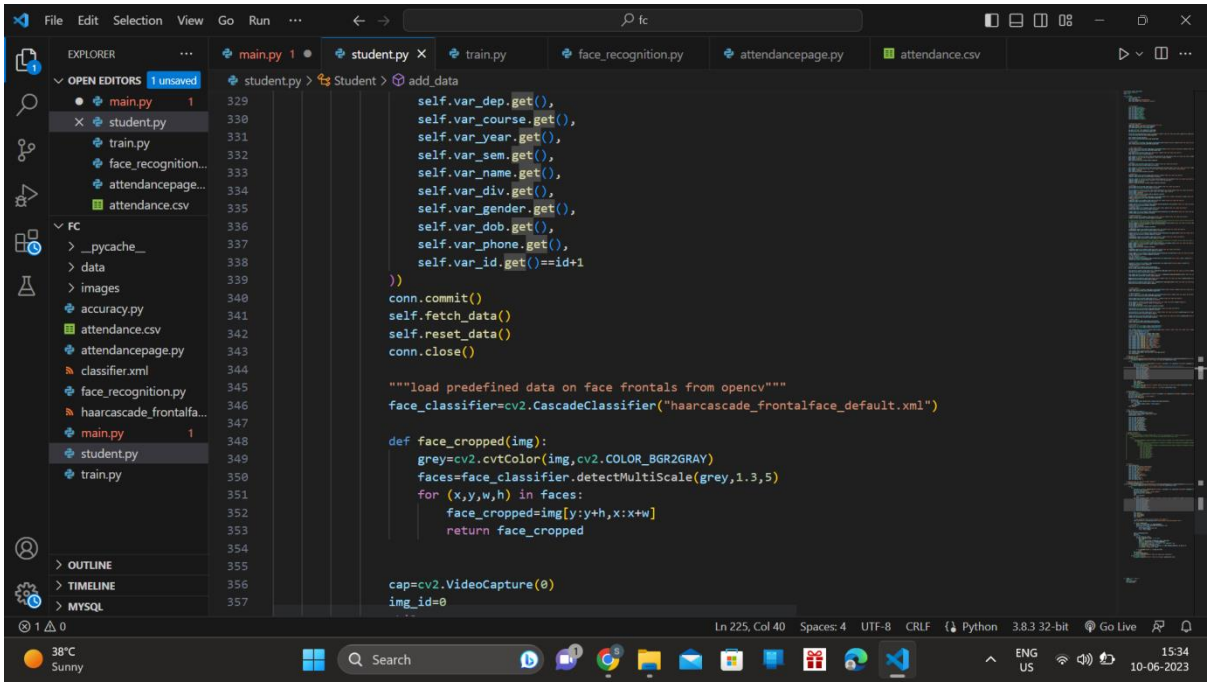

File Edit Selection View Go Run
EXPLORER
OPEN EDITORS 1 unsaved
main.py 1
student.py
train.py
face_recognition...
attendancepage...
attendance.csv
FC
__pycache__
data
images
accuracy.py
attendance.csv
attendancepage.py
classifier.xml
face_recognition.py
haarcascade_frontalfa...
main.py 1
student.py
train.py
OUTLINE
TIMELINE
MYSQL
main.py 1
student.py
train.py
face_recognition.py
attendancepage.py
attendance.csv
student.py > Student > add_data
self.student_table=ttk.Treeview(table_frame,column=("dep","course","year","sem","id","name","div","gender","d
scroll_x.pack(side=BOTTOM,fill=X)
scroll_y.pack(side=RIGHT,fill=Y)
scroll_x.config(command=self.student_table.xview)
scroll_y.config(command=self.student_table.yview)
self.student_table.heading("dep",text="Department")
self.student_table.heading("course",text="Course")
self.student_table.heading("year",text="Year")
self.student_table.heading("sem",text="Semester")
self.student_table.heading("id",text="SudentID")
self.student_table.heading("name",text="Name")
self.student_table.heading("div",text="Division")
self.student_table.heading("dob",text="DOB")
self.student_table.heading("phone",text="Phone no")
self.student_table.heading("gender",text="Gender")
self.student_table["show"]="headings"
self.student_table.pack(fill=BOTH,expand=1)
self.student_table.bind("<ButtonRelease>",self.get_cursor)
self.fetch_data()
"""function for adding data"""
def add_data(self):
if self.var_dep.get()=="Select Department" or self.var_name.get()=="" or self.var_id.get()=="":
messagebox.showerror("Error","All fields are required",parent=self.root)
else:
try:
conn=mysql.connector.connect(host="localhost",username="root",password="shashi8074",database="face_re
Ln 225, Col 40 Spaces: 4 UTF-8 CRLF Python 3.8.3 32-bit Go Live
38°C Sunny
Search
ENG US
15:34 10-06-2023


File Edit Selection View Go Run
EXPLORER
OPEN EDITORS 1 unsaved
main.py 1
student.py
train.py
face_recognition...
attendancepage...
attendance.csv
FC
__pycache__
data
images
accuracy.py
attendance.csv
attendancepage.py
classifier.xml
face_recognition.py
haarcascade_frontalfa...
main.py 1
student.py
train.py
OUTLINE
TIMELINE
MYSQL
main.py 1
student.py
train.py
face_recognition.py
attendancepage.py
attendance.csv
student.py > Student > add_data
else:
try:
conn=mysql.connector.connect(host="localhost",username="root",password="shashi8074",database="face_re
my_cursor=conn.cursor()
my_cursor.execute("insert into student values(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)",(
self.var_dep.get(),
self.var_course.get(),
self.var_year.get(),
self.var_sem.get(),
self.var_id.get(),
self.var_name.get(),
self.var_div.get(),
self.var_gender.get(),
self.var_dob.get(),
self.var_phone.get()
))
conn.commit()
self.fetch_data()
conn.close()
messagebox.showinfo("Success","Student details has been successfully added",parent=self.root)
except Exception as es:
messagebox.showerror("Error",f"Due to : {str(es)}",parent=self.root)
"""fetch data"""
def fetch_data(self):
conn=mysql.connector.connect(host="localhost",username="root",password="shashi8074",database="face_recognitio
my_cursor=conn.cursor()
my_cursor.execute("select * from student")
Ln 225, Col 40 Spaces: 4 UTF-8 CRLF Python 3.8.3 32-bit Go Live
38°C Sunny
Search
ENG US
15:34 10-06-2023

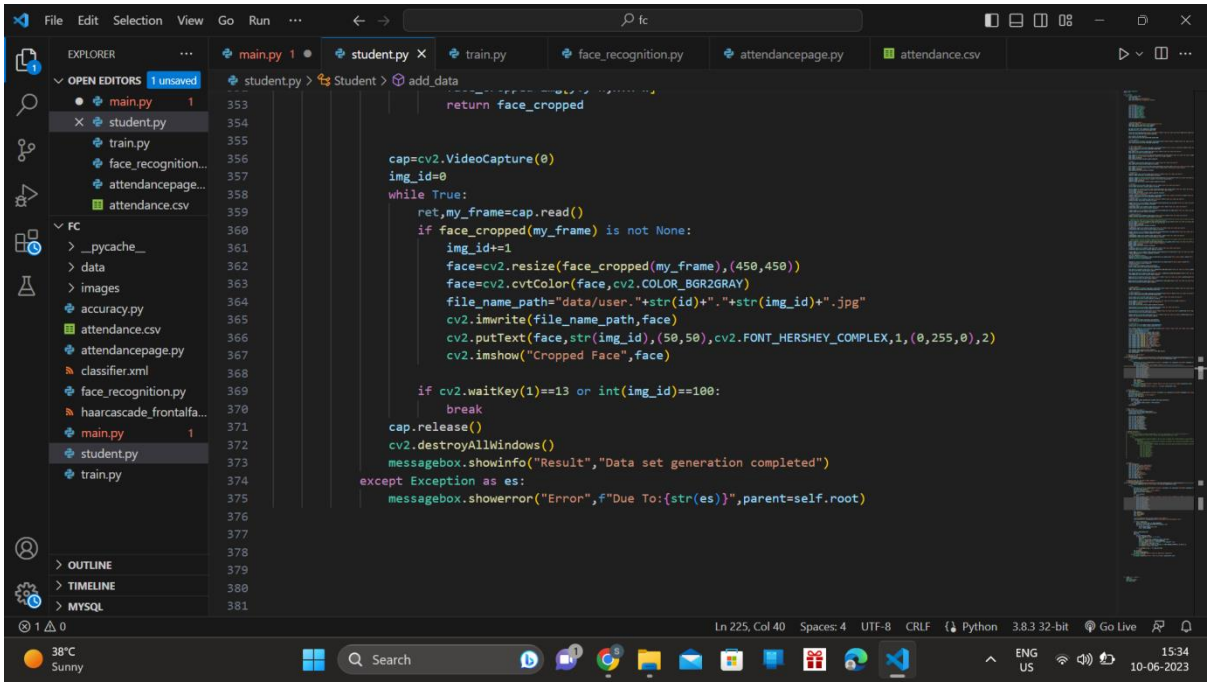

File Edit Selection View Go Run
EXPLORER
OPEN EDITORS 1 unsaved
main.py 1
student.py
train.py
face_recognition...
attendancepage...
attendance.csv
FC
__pycache__
data
images
accuracy.py
attendance.csv
attendancepage.py
classifier.xml
face_recognition.py
haarcascade_frontalfa...
main.py 1
student.py
train.py
OUTLINE
TIMELINE
MYSQL
main.py 1
student.py
train.py
face_recognition.py
attendancepage.py
attendance.csv
student.py > Student > add_data
if len(data)!=0:
self.student_table.delete(*self.student_table.get_children())
for i in data:
self.student_table.insert("",END,values=i)
conn.commit()
conn.close()
"""get cursor"""
def get_cursor(self,event=""):
cursor_focus=self.student_table.focus()
content=self.student_table.item(cursor_focus)
data=content["values"]
self.var_dep.set(data[0]),
self.var_course.set(data[1]),
self.var_year.set(data[2]),
self.var_sem.set(data[3]),
self.var_id.set(data[4]),
self.var_name.set(data[5]),
self.var_div.set(data[6]),
self.var_gender.set(data[7]),
self.var_dob.set(data[8]),
self.var_phone.set(data[9]),
self.var_radio1.set(data[10])
"""update function"""
def update_data(self):
if self.var_dep.get()=="Select Department" or self.var_name.get()=="" or self.var_id.get()=="":
Ln 225, Col 40 Spaces: 4 UTF-8 CRLF Python 3.8.3 32-bit Go Live
38°C Sunny
Search
ENG US
15:34 10-06-2023

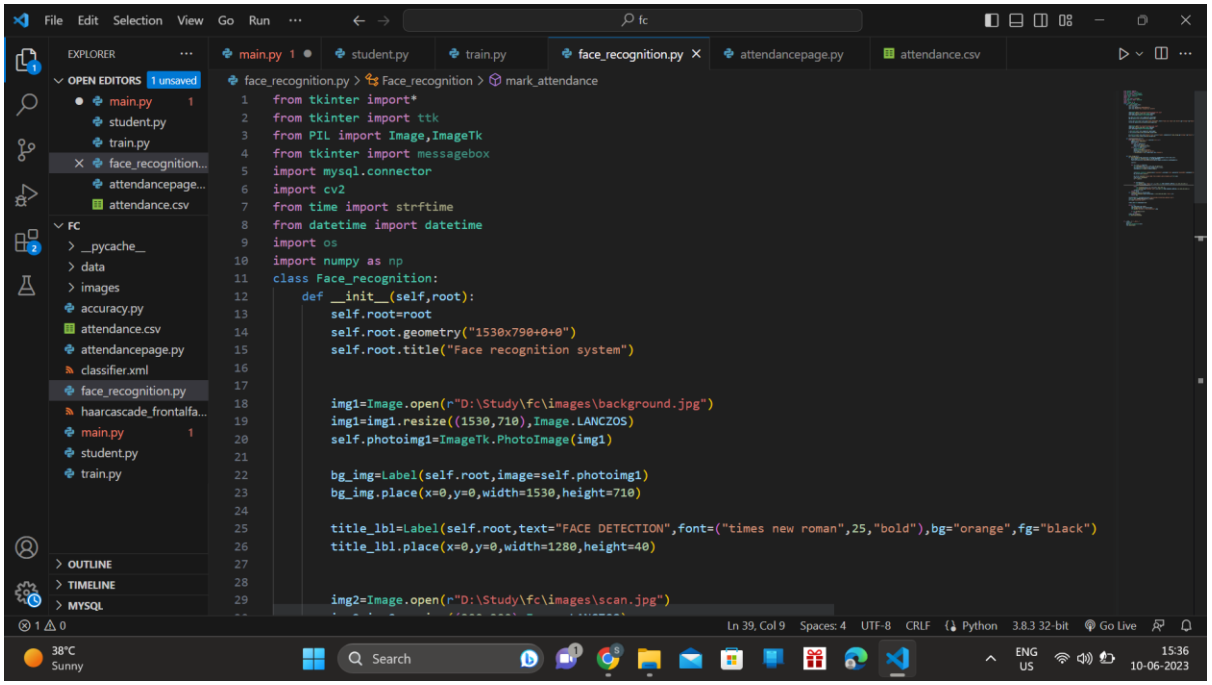


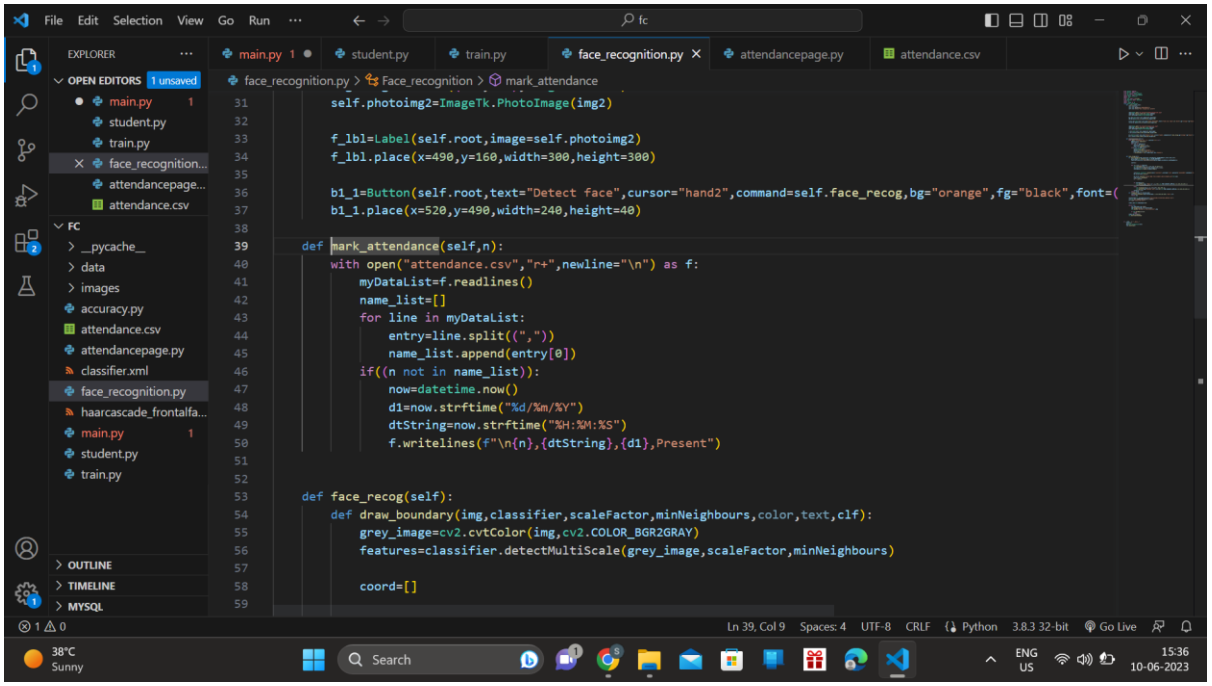
```
"""reset"""
def reset_data(self):
    self.var_dep.set("Select Department")
    self.var_course.set("Select course")
    self.var_year.set("Select year")
    self.var_sem.set("Select semester")
    self.var_id.set("")
    self.var_name.set("")
    self.var_div.set("Select")
    self.var_gender.set("Select")
    self.var_dob.set("")
    self.var_phone.set("")
    self.var_radio1.set("")

"""generate data set and take a photo samples"""
def generate_dataset(self):
    if self.var_dep.get()=="Select Department" or self.var_name.get()=="" or self.var_id=="":
        messagebox.showerror("Error","All fields are required",parent=self.root)
    else:
        try:
            conn=mysql.connector.connect(host="localhost",username="root",password="shashi8074",database="face_re
            my_cursor=conn.cursor()
            my_cursor.execute("select * from student")
            myresult=my_cursor.fetchall()
            id=0
            for x in myresult:
                id+=1
```



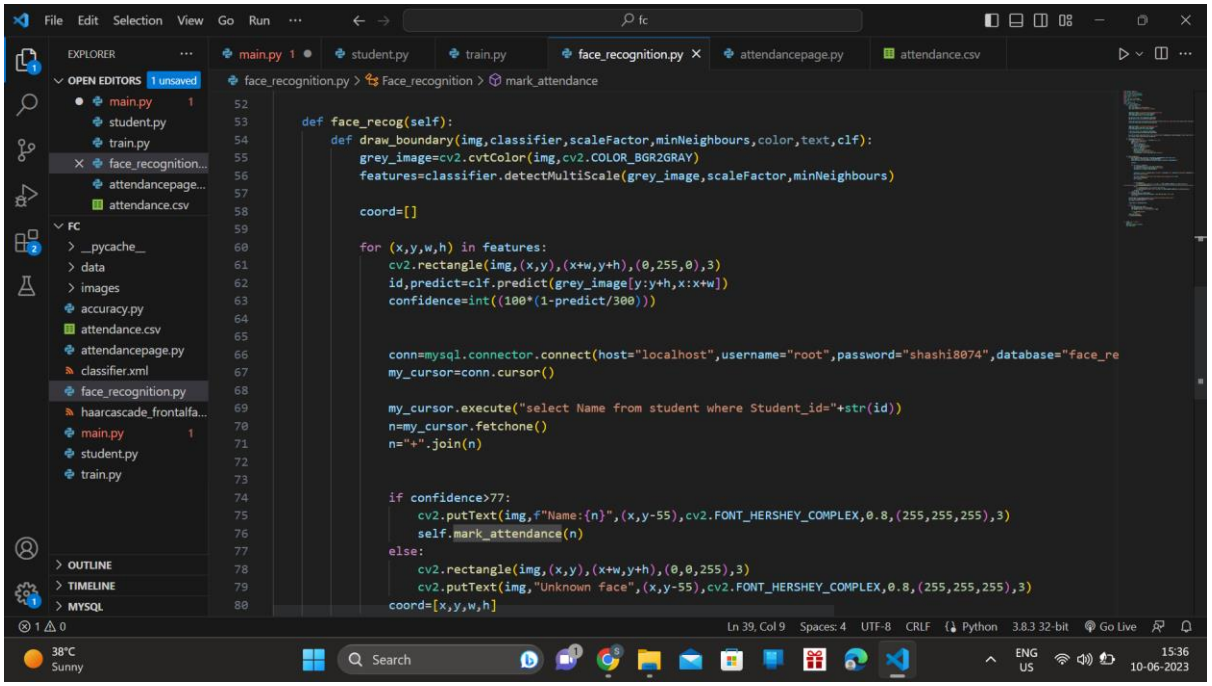
```
                self.var_dep.get(),
                self.var_course.get(),
                self.var_year.get(),
                self.var_sem.get(),
                self.var_name.get(),
                self.var_div.get(),
                self.var_gender.get(),
                self.var_dob.get(),
                self.var_phone.get(),
                self.var_id.get()==id+1
            ))
            conn.commit()
            self.fetch_data()
            self.reset_data()
            conn.close()

            """load predefined data on face frontals from opencv"""
            face_classifier=cv2.CascadeClassifier("haarcascade_frontalface_default.xml")

            def face_cropped(img):
                grey=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
                faces=face_classifier.detectMultiScale(grey,1.3,5)
                for (x,y,w,h) in faces:
                    face_cropped=img[y:y+h,x:x+w]
                    return face_cropped


            cap=cv2.VideoCapture(0)
            img_id=0
```


File Edit Selection View Go Run
fc
EXPLORER
OPEN EDITORS 1 unsaved
main.py 1
student.py
train.py
face_recognition...
attendancepage...
attendance.csv
FC
__pycache__
data
images
accuracy.py
attendance.csv
attendancepage.py
classifier.xml
face_recognition.py
haarcascade_frontalfa...
main.py 1
student.py
train.py
OUTLINE
TIMELINE
MYSQL
main.py 1
student.py
train.py
face_recognition.py
attendancepage.py
attendance.csv
student.py > Student > add_data
return face_cropped
cap=cv2.VideoCapture(0)
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_name_path="data/user."+str(id)+"."+str(img_id)+".jpg"
cv2.imwrite(file_name_path,face)
cv2.putText(face,str(img_id),(50,50),cv2.FONT_HERSHEY_COMPLEX,1,(0,255,0),2)
cv2.imshow("Cropped Face",face)
if cv2.waitKey(1)==13 or int(img_id)==100:
break
cap.release()
cv2.destroyAllWindows()
messagebox.showinfo("Result","Data set generation completed")
except Exception as es:
messagebox.showerror("Error",f"Due To:{str(es)}",parent=self.root)
Ln 225, Col 40 Spaces: 4 UTF-8 CRLF Python 3.8.3 32-bit Go Live
38°C Sunny
Search
ENG US
15:34 10-06-2023

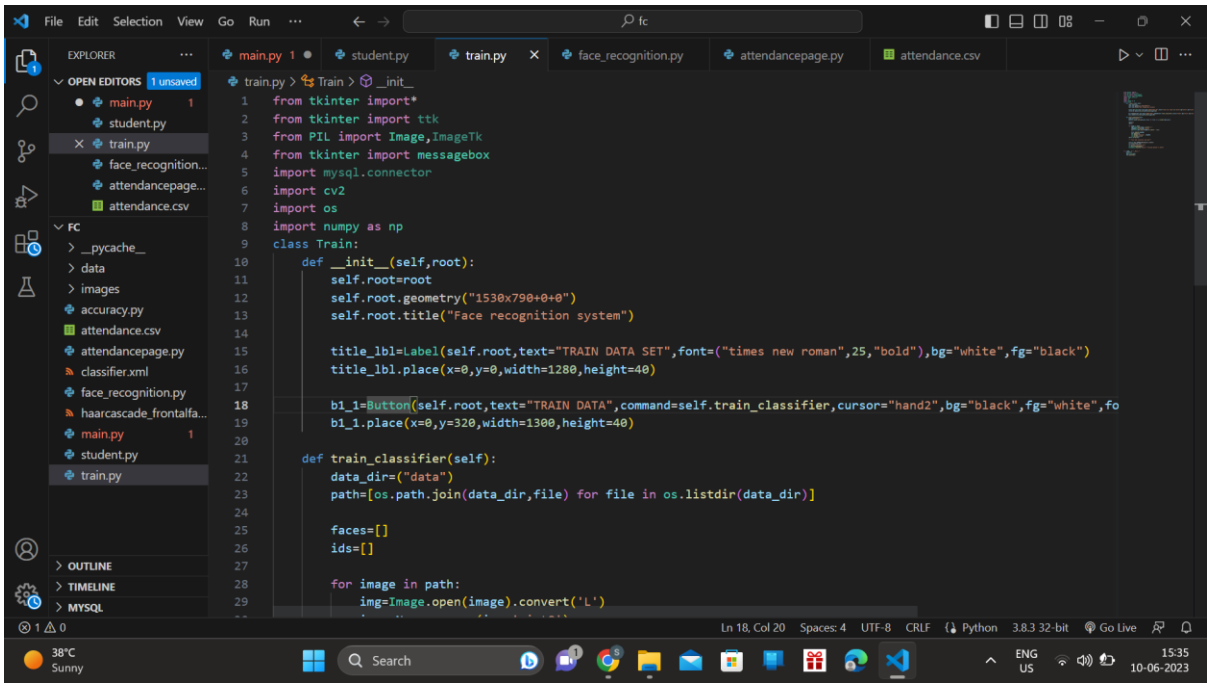

File Edit Selection View Go Run
fc
EXPLORER
OPEN EDITORS 1 unsaved
main.py 1
student.py
train.py
face_recognition...
attendancepage...
attendance.csv
FC
__pycache__
data
images
accuracy.py
attendance.csv
attendancepage.py
classifier.xml
face_recognition.py
haarcascade_frontalfa...
main.py 1
student.py
train.py
OUTLINE
TIMELINE
MYSQL
main.py 1
student.py
train.py
face_recognition.py
attendancepage.py
attendance.csv
face_recognition.py > Face_recognition > mark_attendance
from tkinter import*
from tkinter import ttk
from PIL import Image,ImageTk
from tkinter import messagebox
import mysql.connector
import cv2
from time import strftime
from datetime import datetime
import os
import numpy as np
class Face_recognition:
def __init__(self,root):
self.root=root
self.root.geometry("1530x790+0+0")
self.root.title("Face recognition system")
img1=Image.open(r"D:\Study\fc\images\background.jpg")
img1=img1.resize((1530,710),Image.LANCZOS)
self.photoimg1=ImageTk.PhotoImage(img1)
bg_img=Label(self.root,image=self.photoimg1)
bg_img.place(x=0,y=0,width=1530,height=710)
title_lbl=Label(self.root,text="FACE DETECTION",font=("times new roman",25,"bold"),bg="orange",fg="black")
title_lbl.place(x=0,y=0,width=1280,height=40)
img2=Image.open(r"D:\Study\fc\images\scan.jpg")
Ln 39, Col 9 Spaces: 4 UTF-8 CRLF Python 3.8.3 32-bit Go Live
38°C Sunny
Search
ENG US
15:36 10-06-2023

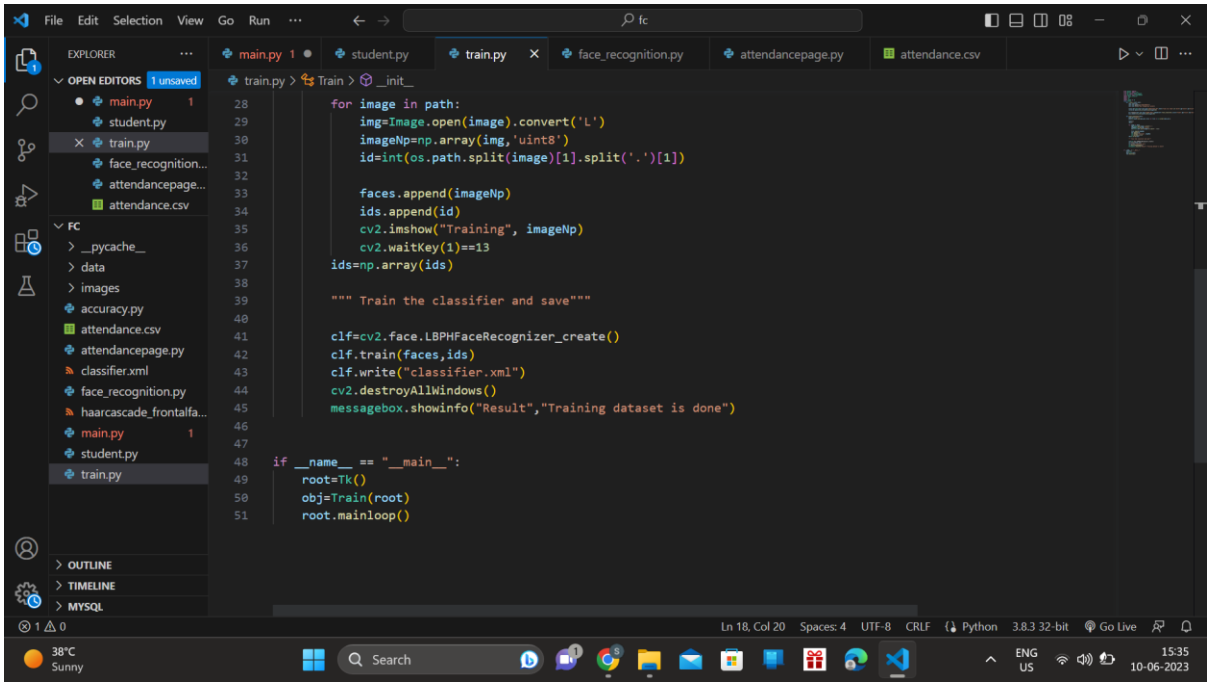


```
        self.photoimg2=ImageTk.PhotoImage(img2)

        f_lbl=Label(self.root,image=self.photoimg2)
        f_lbl.place(x=490,y=160,width=300,height=300)

        b1_1=Button(self.root,text="Detect face",cursor="hand2",command=self.face_recog,bg="orange",fg="black",font=(
        b1_1.place(x=520,y=490,width=240,height=40)

    def mark_attendance(self,n):
        with open("attendance.csv","r+",newline="\n") as f:
            myDataList=f.readlines()
            name_list=[]
            for line in myDataList:
                entry=line.split((","))
                name_list.append(entry[0])
            if((n not in name_list)):
                now=datetime.now()
                d1=now.strftime("%d/%m/%Y")
                dtString=now.strftime("%H:%M:%S")
                f.writelines(f"\n{n},{dtString},{d1},Present")


    def face_recog(self):
        def draw_boundary(img,classifier,scaleFactor,minNeighbours,color,text,clf):
            grey_image=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
            features=classifier.detectMultiScale(grey_image,scaleFactor,minNeighbours)

            coord=[]
```



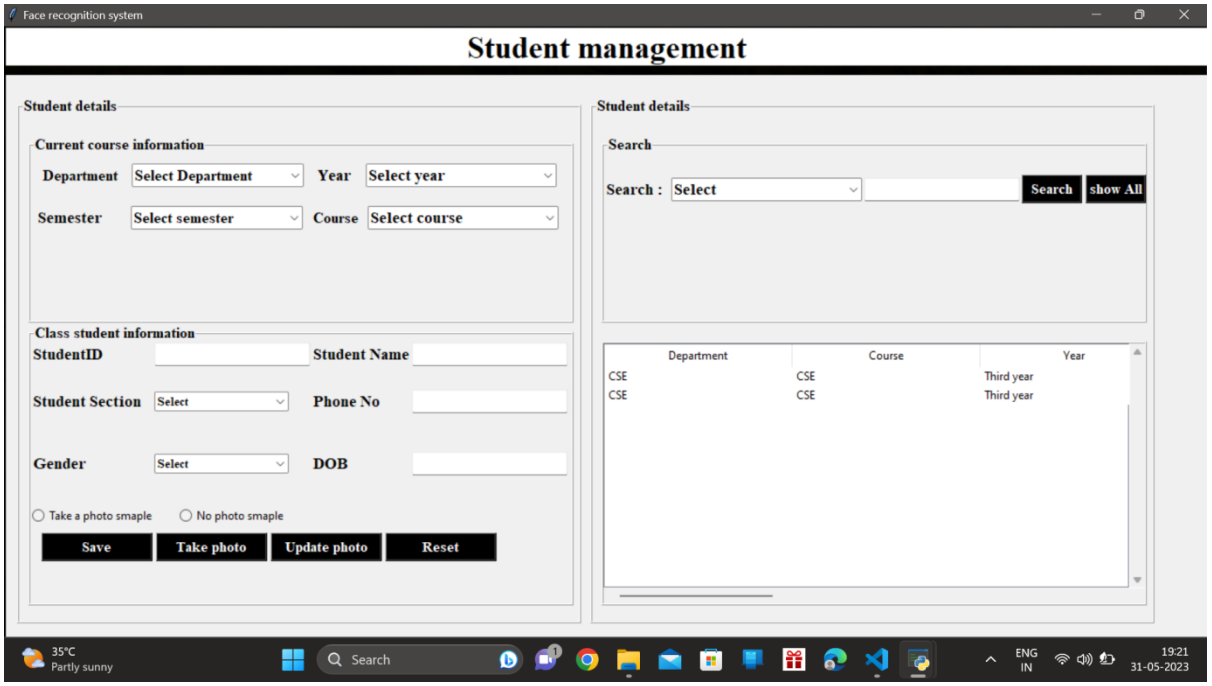
```
    def face_recog(self):
        def draw_boundary(img,classifier,scaleFactor,minNeighbours,color,text,clf):
            grey_image=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
            features=classifier.detectMultiScale(grey_image,scaleFactor,minNeighbours)

            coord=[]

            for (x,y,w,h) in features:
                cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),3)
                id,predict=clf.predict(grey_image[y:y+h,x:x+w])
                confidence=int((100*(1-predict/300)))


                conn=mysql.connector.connect(host="localhost",username="root",password="shashi8074",database="face_re
                my_cursor=conn.cursor()

                my_cursor.execute("select Name from student where Student_id="+str(id))
                n=my_cursor.fetchone()
                n="+".join(n)


                if confidence>77:
                    cv2.putText(img,f"Name:{n}",(x,y-55),cv2.FONT_HERSHEY_COMPLEX,0.8,(255,255,255),3)
                    self.mark_attendance(n)
                else:
                    cv2.rectangle(img,(x,y),(x+w,y+h),(0,0,255),3)
                    cv2.putText(img,"Unknown face",(x,y-55),cv2.FONT_HERSHEY_COMPLEX,0.8,(255,255,255),3)
                coord=[x,y,w,h]
```

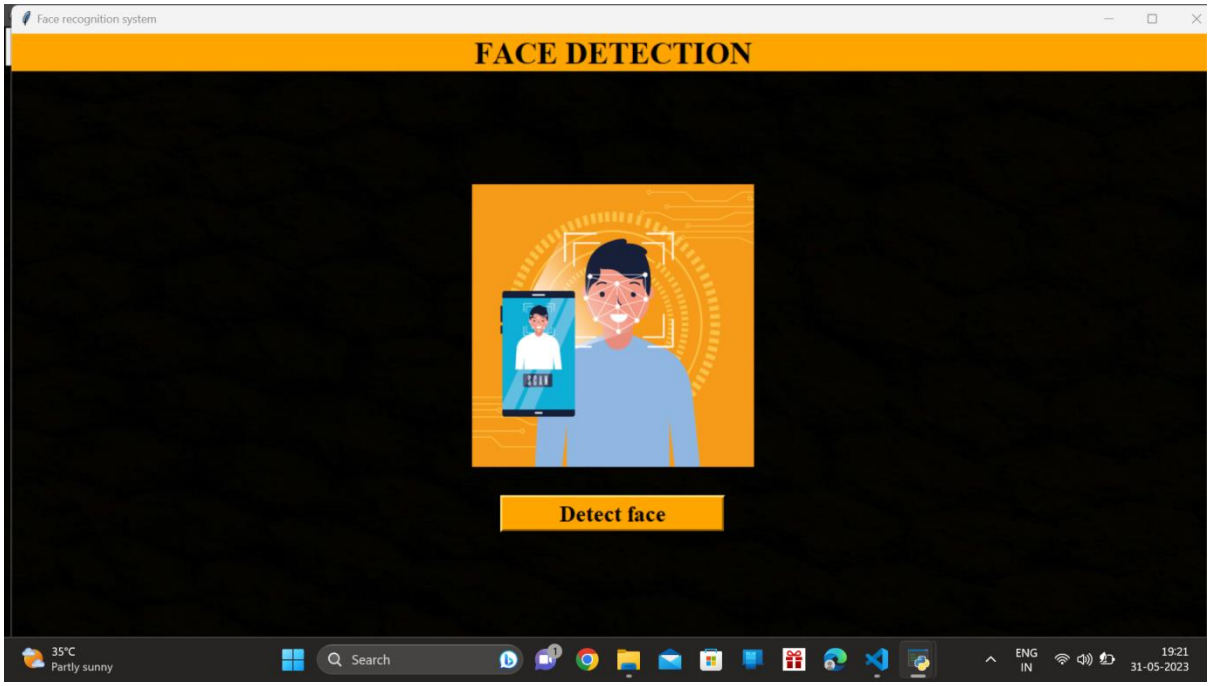

File Edit Selection View Go Run
EXPLORER
OPEN EDITORS 1 unsaved
main.py 1
student.py
train.py
face_recognition...
attendancepage...
attendance.csv
FC
__pycache__
data
images
accuracy.py
attendance.csv
attendancepage.py
classifier.xml
face_recognition.py
haarcascade_frontalfa...
main.py 1
student.py
train.py
OUTLINE
TIMELINE
MYSQL
main.py 1
student.py
train.py
face_recognition.py
attendancepage.py
attendance.csv
train.py > Train > __init__
from tkinter import*
from tkinter import ttk
from PIL import Image,ImageTk
from tkinter import messagebox
import mysql.connector
import cv2
import os
import numpy as np
class Train:
def __init__(self,root):
self.root=root
self.root.geometry("1530x790+0+0")
self.root.title("Face recognition system")
title_lbl=Label(self.root,text="TRAIN DATA SET",font=("times new roman",25,"bold"),bg="white",fg="black")
title_lbl.place(x=0,y=0,width=1280,height=40)
b1_1=Button(self.root,text="TRAIN DATA",command=self.train_classifier,cursor="hand2",bg="black",fg="white",fo
b1_1.place(x=0,y=320,width=1300,height=40)
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')
Ln 18, Col 20 Spaces: 4 UTF-8 CRLF Python 3.8.3 32-bit Go Live
38°C Sunny
Search
ENG US
15:35 10-06-2023

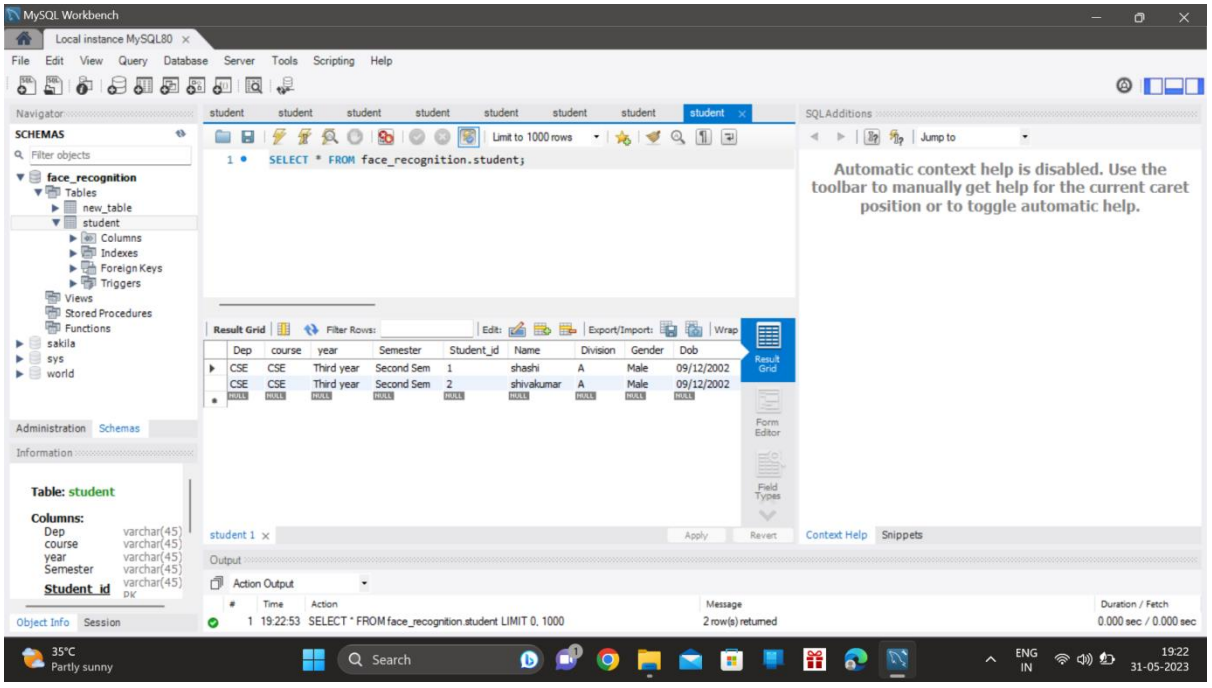

File Edit Selection View Go Run
EXPLORER
OPEN EDITORS 1 unsaved
main.py 1
student.py
train.py
face_recognition...
attendancepage...
attendance.csv
FC
__pycache__
data
images
accuracy.py
attendance.csv
attendancepage.py
classifier.xml
face_recognition.py
haarcascade_frontalfa...
main.py 1
student.py
train.py
OUTLINE
TIMELINE
MYSQL
main.py 1
student.py
train.py
face_recognition.py
attendancepage.py
attendance.csv
train.py > Train > __init__
for image in path:
img=Image.open(image).convert('L')
imageNp=np.array(img,'uint8')
id=int(os.path.split(image)[1].split('.')[1])
faces.append(imageNp)
ids.append(id)
cv2.imshow("Training", imageNp)
cv2.waitKey(1)==13
ids=np.array(ids)
""" Train the classifier and save"""
clf=cv2.face.LBPHFaceRecognizer_create()
clf.train(faces,ids)
clf.write("classifier.xml")
cv2.destroyAllWindows()
messagebox.showinfo("Result","Training dataset is done")
if __name__ == "__main__":
root=Tk()
obj=Train(root)
root.mainloop()
Ln 18, Col 20 Spaces: 4 UTF-8 CRLF Python 3.8.3 32-bit Go Live
38°C Sunny
Search
ENG US
15:35 10-06-2023

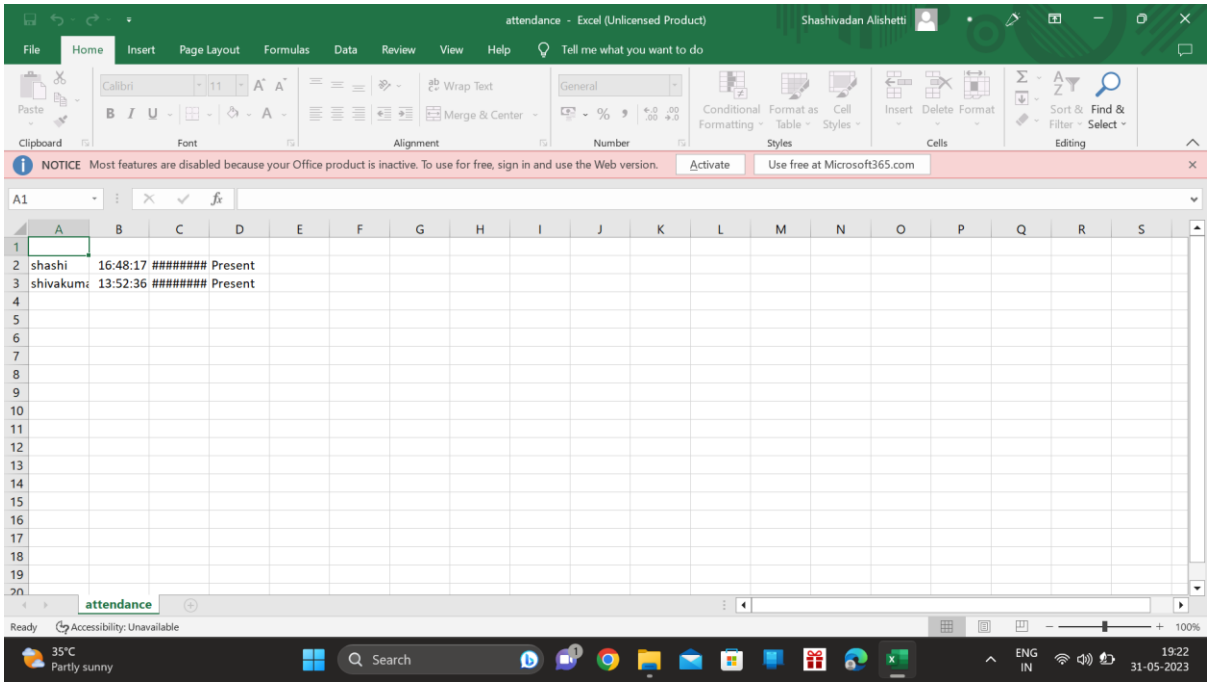
5.2 Result


Face recognition system
FACE RECOGNITION SYSTEM
Student Details
Detect Face
Train data
Attendance
35°C
Partly sunny
Search
ENG
IN
19:21
31-05-2023


Face recognition system
Student management
Student details
Current course information
Department
Select Department
Year
Select year
Semester
Select semester
Course
Select course
Class student information
StudentID
Student Name
Student Section
Select
Phone No
Gender
Select
DOB
Take a photo smaple
No photo smaple
Save
Take photo
Update photo
Reset
Student details
Search
Search :
Select
Search
show All
Department
Course
Year
CSE
CSE
Third year
CSE
CSE
Third year
35°C
Partly sunny
Search
ENG
IN
19:21
31-05-2023


Face recognition system
FACE DETECTION
Detect face
35°C
Partly sunny
Search
ENG
IN
19:21
31-05-2023


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SELECT * FROM face_recognition.student;
face_recognition
Tables
new_table
student
Columns
Indexes
Foreign Keys
Triggers
Views
Stored Procedures
Functions
sakila
sys
world
Dep course year Semester Student_id Name Division Gender Dob
CSE CSE Third year Second Sem 1 shashi A Male 09/12/2002
CSE CSE Third year Second Sem 2 shivakumar A Male 09/12/2002
Table: student
Columns:
Dep varchar(45)
course varchar(45)
year varchar(45)
Semester varchar(45)
Student_id varchar(45)
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Action Output
1 19:22:53 SELECT * FROM face_recognition.student LIMIT 0, 1000
2 row(s) returned
0.000 sec / 0.000 sec
35°C
Partly sunny
Search
ENG
IN
19:22
31-05-2023


attendance - Excel (Unlicensed Product)
Shashivadan Alishetti
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Help
Tell me what you want to do
NOTICE Most features are disabled because your Office product is inactive. To use for free, sign in and use the Web version.
Activate
Use free at Microsoft365.com
A1
shashi 16:48:17 ######## Present
shivakuma 13:52:36 ######## Present
attendance
Ready
Accessibility: Unavailable
35°C
Partly sunny
Search
ENG
IN
19:22
31-05-2023

CHAPTER 6

TESTING

Test Case ID	Input 1	Expected output	Actual output	Status Pass/Fail
01	Real image of person 1	Genuine	Genuine	Pass
02	Fake image of person 1	Forged	Forged	Pass
03	Real image of person 2	Genuine	Genuine	Pass
04	Real image of person 3	Genuine	Genuine	Pass
05	Fake image of person 2	Forged	Forged	Pass
06	Fake image of person 3	Forged	Forged	Pass
07	Real image of person 4	Genuine	Genuine	Pass
08	Real image of person 5	Genuine	Genuine	Pass
09	Fake image of person 4	Forged	Forged	Pass
10	Fake image of person 5	Forged	Forged	Pass

Table 6.1 Testing

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

Conclusion:

The face recognition attendance system using Haar Cascade Frontal Face algorithm and LBPH algorithm offers a reliable and efficient solution for attendance management in various organizations. It provides accurate face detection, recognition, and attendance marking capabilities, ensuring an automated and streamlined process. The system has demonstrated its effectiveness in accurately identifying enrolled employees, marking their attendance in real-time, and maintaining comprehensive attendance records. It enhances security, eliminates manual attendance processes, and reduces administrative overhead.

Future Scope: While the current implementation of the face recognition attendance system using Haar Cascade Frontal Face algorithm and LBPH algorithm is promising, there are several potential areas for future improvement and expansion

1. Performance Enhancement: Further optimizations can be explored to improve the system's performance, such as parallel processing, hardware acceleration, or algorithmic improvements to speed up face detection and recognition.

2. Advanced Recognition Techniques: The system can be enhanced by integrating more advanced face recognition algorithms, such as deep learning-based approaches like Convolutional Neural Networks (CNNs) or pretrained models like VGGFace or FaceNet, to improve accuracy and robustness.

3. Multi-Factor Authentication: Integration of additional authentication factors, such as fingerprint or iris recognition, can strengthen the security of the attendance system and prevent impersonation or fraudulent activities.

4. Mobile Integration: Developing mobile applications can provide employees with the convenience of marking attendance using their smartphones, enhancing accessibility and usability.

5. Real-Time Analytics and Insights: Implementing real-time analytics and reporting features can enable administrators to gain valuable insights into attendance patterns, identify trends, and generate customized reports for better decision-making.

6. Integration with Access Control Systems: Integrating the attendance system with access control systems can facilitate seamless entry and exit management based on employee attendance records.

REFERENCES

"DeepFace: Closing the Gap to Human-Level Performance in Face Verification" by Yaniv Taigman, Ming Yang, Marc'AurelioRanzato, and Lior Wolf.

"DeepID3: Face Recognition with Very Deep Neural Networks" by Yi Sun, Xiaogang Wang, and Xiaoou Tang.

"VGGFace: Deep Face Recognition" by Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman.

"FaceNet: A Unified Embedding for Face Recognition and Clustering" by Florian Schroff, Dmitry Kalenichenko, and James Philbin.

"Face recognition with OpenCV, Python, and deep learning" by Adrian Rosebrock on PyImageSearch.

"Face Detection and Recognition on Custom Images using OpenCV and Python" by Prateek Joshi on Analytics Vidhya.

"Real-time Face Recognition: An End-to-End Project" by Satya Mallick on Learn OpenCV.

"Face Recognition: From Scratch to Hatch" by Vinay Prabhu on Towards Data Science.

"Automated Attendance Management System using Face Recognition" by Amrita Bagga, Mandeep Kaur, and Harsimran Kaur.

"Face Recognition-based Attendance System" by Pooja Bairagi, Trusha Y. Shedge, and Priyanka D. Chandanshive.

"Automated Attendance System Using Face Recognition" by S. Harish and S. Meghana.

"Real-Time Student Attendance Monitoring System using Face Recognition" by Elsayed Abdel-Raheem, Mohamed I. Soliman, and Hoda M. O. Mokhtar.