Project Title

Face Recognition Attendance System

Semester project

2023-2027

BS Artificial Intelligence

Department of Software Engineering

Faculty of Computer Science & Information Technology

The Superior University, Lahore

FALL 2024

Type (Nature of project)	[ ] **D**evelopment [ ] **R**esearch [ ] **R&D**
Area of specialization	Machine Learning

Project Group Members

Sr.#	Reg. #	Student Name	Email ID	*Signature
1	SU92-BSAIM-F23-070	M. Abdullah Kashif	abdullahkashifn@gmail.com	

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Table of Contents

Figure's

Chapter 1

Introduction

Write Introduction about your project

Introduction to Face Recognition Attendance System

The "Face Recognition Attendance System" project is an advanced application of artificial intelligence and computer vision designed to streamline and automate attendance management. By utilizing facial recognition technology, this system provides an efficient, contactless, and secure method to register and verify individuals' attendance. This solution is ideal for organizations, educational institutions, and workplaces to enhance record-keeping accuracy and minimize manual errors.

Key

1. **Face Registration**:
    - Allows individuals to register their facial data along with their name and roll number.
    - Ensures unique identification by detecting duplicate entries during registration.
2. **Facial Recognition**:
    - Employs advanced facial recognition models to identify individuals from an image.
    - Uses local binary patterns histogram (LBPH) and Haar cascades for effective face detection and recognition.
3. **Attendance Logging**:
    - Automatically marks attendance in a structured CSV file with details such as name, roll number, date, and time.

Benefits:

This project exemplifies the practical application of AI in daily operations by automating routine tasks, ensuring secure authentication, and providing a scalable solution for attendance management. It demonstrates how technology can simplify traditional processes while fostering a seamless and user-friendly experience.

Chapter 2

Tool & Technology

Write tool and technology which you have used

Software and AI Models for Face Recognition Attendance System

Programming Languages:

- **Python**: The core language for implementing facial recognition, data processing, and attendance management.

AI and Machine Learning Frameworks:

1. **Image Processing:**
    - OpenCV: For face detection and recognition tasks.
    - NumPy: For efficient array manipulations during image and data handling.
    - OS: For interacting with the file system to manage student data and attendance records.
2. **Face Recognition Frameworks:**
    - **LBPH (Local Binary Patterns Histogram):** Used for facial recognition due to its simplicity and effectiveness for smaller datasets.
3. **Data Management:**
    - **Pandas**: For creating, reading, and managing attendance logs in CSV format.

Integration Technologies:

- **Face Recognition Workflow:**
    - Uses Haar cascades (pre-trained models in OpenCV) to detect faces in images.
    - LBPH recognizer processes and matches faces to registered student data.

Development Tools:

- **IDE and Editors:**
    - Python-compatible tools like PyCharm, VS Code, or Jupyter Notebook for writing and debugging the code.

Device name DESKTOP-PMO8KCT

Processor Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz 2.50 GHz

Installed RAM 8.00 GB (7.88 GB usable)

Device ID CDE9CC9B-81DE-4E64-9F85-8F9B7C7FABE5

Product ID 00331-12273-86382-AB517

System type 64-bit operating system, x64-based processor

Edition Windows 10 Pro

Version 22H2

Installed on 8/25/2024

OS build 19045.5198

Experience Windows Feature Experience Pack 1000.19060.1000.0

Chapter 3

Implementation Code

main.py > ...

```python
import cv2
import os
import numpy as np
import pandas as pd
from datetime import datetime
from tkinter import *
from tkinter import filedialog, messagebox

#Images
students_dir = 'students_data'
os.makedirs(students_dir, exist_ok=True)

#CSV
attendance = 'attendance.csv'
frecognize = cv2.face.LBPHFaceRecognizer_create()
fcascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')

def is_registered(rollno):
    stud_file = os.listdir(students_dir)
    for file in stud_file:
        if file.split('_')[1].split('.')[0] == rollno.zfill(3):
            return True
    return False

def register_stud(name, rollno, image):
    if is_registered(rollno):
        messagebox.showerror("Error", f"Student with Roll No: {rollno} is already registered.")
        return False
    image = cv2.imread(image)
    if image is None:
```

```
def register_stud(name, rollno, image):

    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    face = fcascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5)
    if len(face) ==
        messagebox.s                                    ced. Try again.")
        return False

# Check registered s
    if len(os.listdi
        registered_f
        labels = []
        for file in
            stud_img = cv2.imread(os.path.join(students_dir, file), 0)
            if stud_img is not None:
                registered_faces.append(stud_img)
                labels.append(int(file.split('_')[1].split('.')[0]))

        frecognize.train(registered_faces, np.array(labels))

# Check if registered
        for (x, y, w, h) in face:
            face_region = gray[y:y + h, x:x + w]
            try:
                label, confidence = frecognize.predict(face_region)
                if confidence < 50:
                    messagebox.showerror("Error", f"This face is already registered with a different Roll
                    return False
```

```
(method) def detectMultiScale(
    image: MatLike,
    scaleFactor: float = ...,
    minNeighbors: int = ...,
    flags: int = ...,
    minSize: Size = ...,
    maxSize: Size = ...
) -> Sequence[Rect]
```

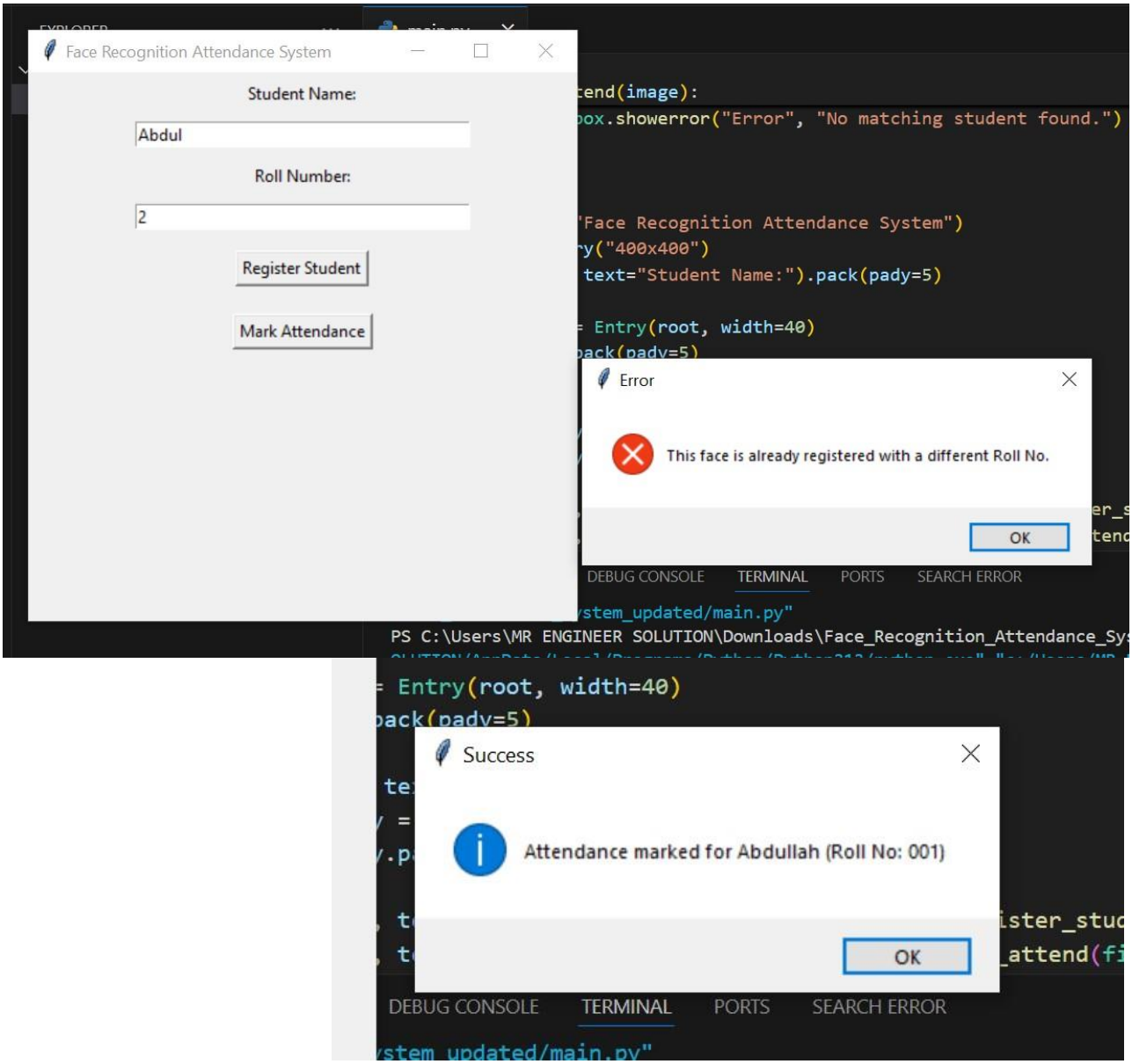
```
def register_stud(name, rollno, image):
                print(f"Error during face recognition: {e}")
                pass

# Save faces        (variable) w: int
    for (x, y, w, h) in face:
        face_region = gray[y:y + h, x:x + w]
        stud_file = os.path.join(students_dir, f"{name}_{rollno.zfill(3)}.jpg")
        cv2.imwrite(stud_file, face_region)
        messagebox.showinfo("Success", f"Student {name} (Roll No: {rollno}) registered successfully.")
        return True

    return False


def mark_attend(image):
    global attendance
    # Time limit 9:00 AM
    now = datetime.now()
    curr_time = now.strftime("%H:%M")
    # if curr_time > "09:00":
    #     messagebox.showerror("Late", "You are late! Attendance cannot be marked after 9:00 AM.")
    #     return

# Create CSV
    if not os.path.exists(attendance):
        pd.DataFrame(columns=['Name', 'Roll No', 'Date', 'Time']).to_csv(attendance, index=False)

    image = cv2.imread(image)
    if image is None:
```

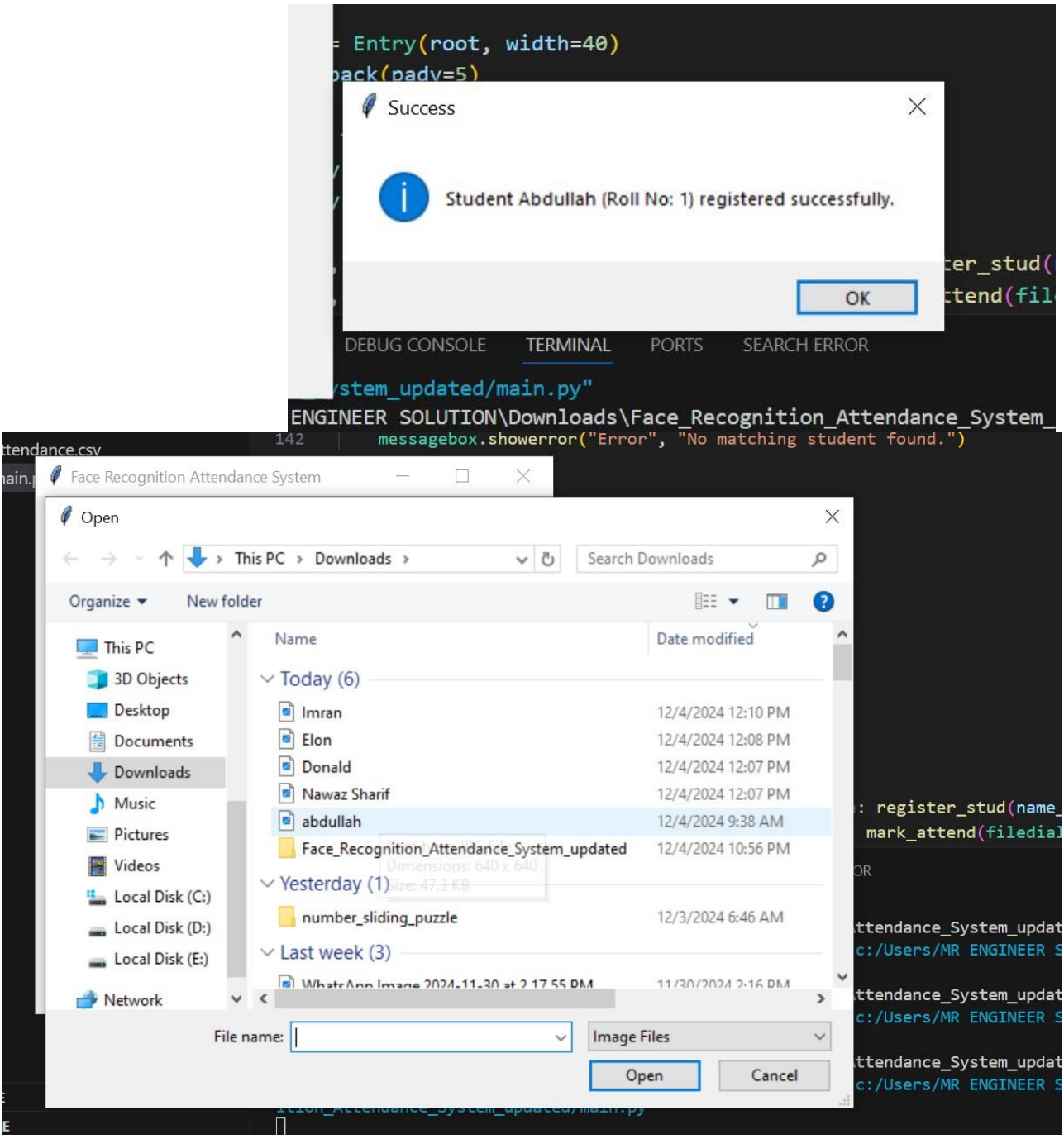
```
def mark_attend(image):
        return

    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    face = fcascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5)
    if len(face) == 0:
        messagebox.showerror("Error", "No face detected. Try again with a clearer image.")
        return

    registered_faces = []
    labels = []
    stud_file = []

    for file in os.listdir(students_dir):
        stud_img = cv2.imread(os.path.join(students_dir, file), 0)
        if stud_img is not None:
            registered_faces.append(stud_img)
            labels.append(int(file.split('_')[1].split('.')[0]))
            stud_file.append(file)

    if len(registered_faces) == 0:
        messagebox.showerror("Error", "No students registered yet.")
        return

# Training
    frecognize.train(registered_faces, np.array(labels))

# Check faces
    for (x, y, w, h) in face:
        face_region = gray[y:y + h, x:x + w]
```

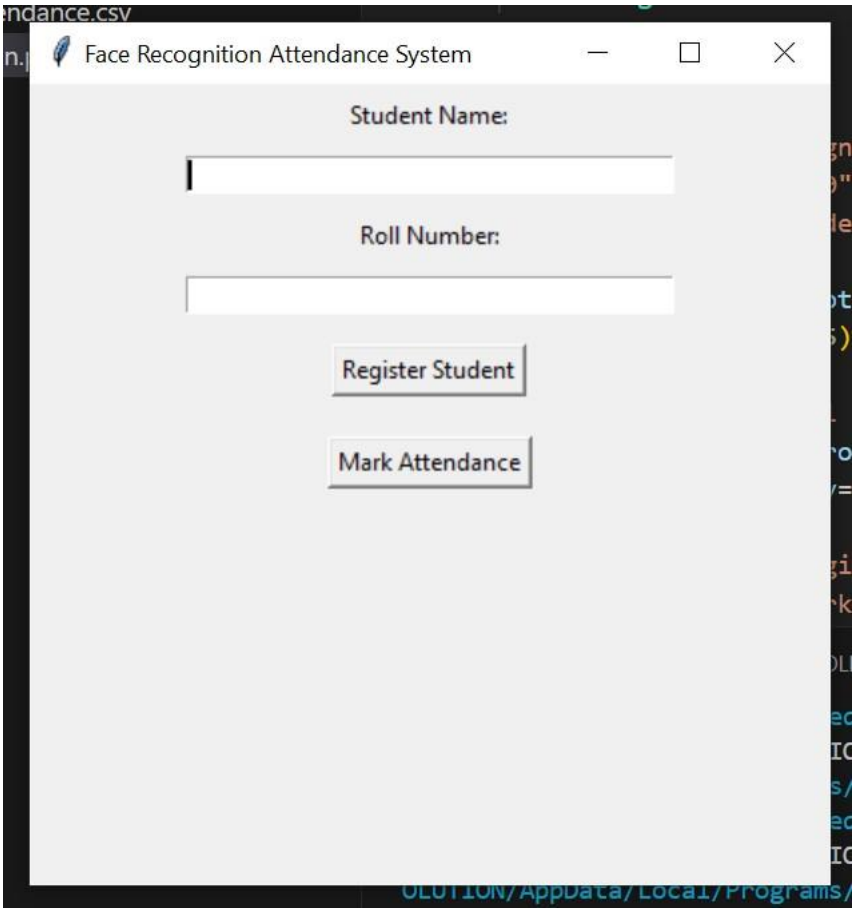
```
root = Tk()
root.title("Face Recognition Attendance System")
root.geometry("400x400")
Label(root, text="Student Name:").pack(pady=5)

name_entry = Entry(root, width=40)
name_entry.pack(pady=5)

Label(root, text="Roll Number:").pack(pady=5)
rollno_entry = Entry(root, width=40)
rollno_entry.pack(pady=5)

Button(root, text="Register Student", command=lambda: register_stud(name_entry.get(), rollno_entry.get(),
Button(root, text="Mark Attendance", command=lambda: mark_attend(filedialog.askopenfilename(filetypes=[("I
root.mainloop()
```

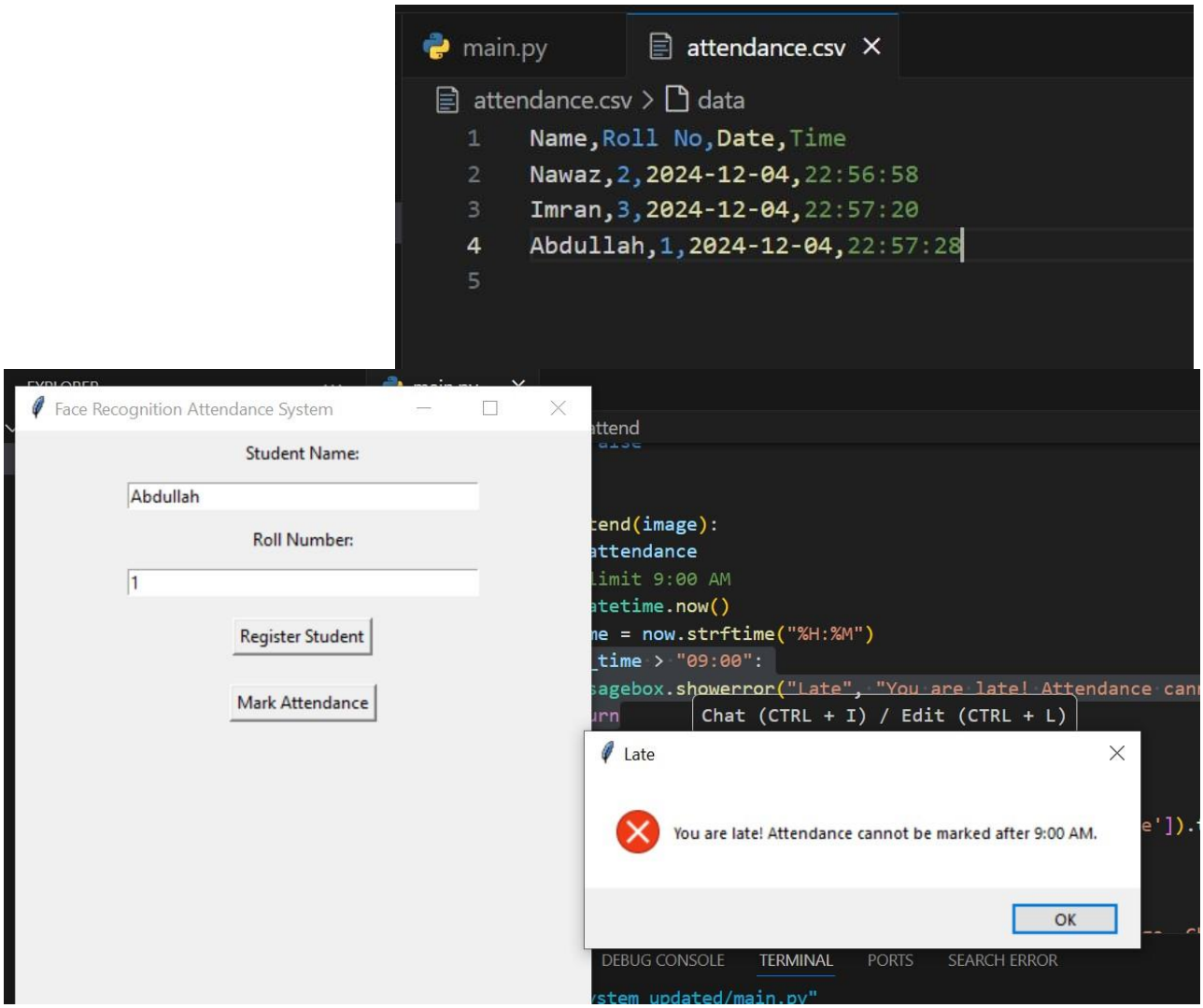
Chapter 4

Result




Entry(root, width=40)
Success
Student Abdullah (Roll No: 1) registered successfully.
OK
DEBUG CONSOLE
TERMINAL
PORTS
SEARCH ERROR
ENGINEER SOLUTION\Downloads\Face_Recognition_Attendance_System_
messagebox.showerror("Error", "No matching student found.")
Face Recognition Attendance System
Open
This PC › Downloads ›
Search Downloads
Organize
New folder
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Local Disk (D:)
Local Disk (E:)
Network
Name
Date modified
Today (6)
Imran 12/4/2024 12:10 PM
Elon 12/4/2024 12:08 PM
Donald 12/4/2024 12:07 PM
Nawaz Sharif 12/4/2024 12:07 PM
abdullah 12/4/2024 9:38 AM
Face_Recognition_Attendance_System_updated 12/4/2024 10:56 PM
Yesterday (1)
number_sliding_puzzle 12/3/2024 6:46 AM
Last week (3)
File name:
Image Files
Open
Cancel


Face Recognition Attendance System
Student Name:
Roll Number:
Register Student
Mark Attendance



What I Learned from This Project

This project provided an enriching learning experience and practical knowledge in several domains of computer vision, machine learning, and software development. Below are the key takeaways from the project:

1. Facial Recognition and Machine Learning

- **Training Face Recognition Models:**

  - Learned how to use LBPH (Local Binary Patterns Histogram) for face recognition, a lightweight and effective model for small datasets.
  - Gained insights into training and validating models, recognizing the importance of confidence thresholds for accurate recognition.
- **Data Handling and Preprocessing:**
  - Understood the role of data preprocessing, including grayscale conversion and facial region extraction, for improving recognition accuracy.

2. Image Processing Techniques

- **Face Detection:**
  - Worked with OpenCV's Haar cascades for real-time face detection, learning its practical applications and limitations.
  - Gained familiarity with scaling, region cropping, and histogram normalization to optimize detection accuracy under different lighting conditions.
- **File and Image Management:**
  - Learned to manage student data and face images systematically using Python's OS and NumPy libraries.

3. Real-Time Attendance Management

- **Real-Time Processing:**
  - Acquired knowledge about capturing, processing, and analyzing images in real-time using webcams or saved images.
  - Understood the integration of detection and recognition workflows for seamless performance in real-time systems.
- **Attendance Logging:**
  - Learned how to utilize Pandas for effective management of attendance records, including creating, reading, and updating CSV files.

4. System Integration and Extensions

- Explored the possibility of incorporating Text-to-Speech (TTS) for audio feedback, which would enhance the user experience.
- Understood how to design a user interface using Python's Tkinter to make the system interactive and user-friendly.

How I Will Use These Skills in Future Projects

1. Handling and Preprocessing Images

- **Improved Image Preparation:**
  - I will apply the skills I've gained in image resizing, normalization, and enhancement to ensure images are consistently prepared for analysis in machine learning models.
  - Advanced techniques, such as converting images to tensors and handling variations in lighting and resolution, will be prioritized to improve the accuracy and robustness of future models.
- **Optimized Data Pipelines:**
  - I plan to create efficient workflows for managing large datasets, ensuring seamless preprocessing for projects requiring real-time analysis or large-scale training.

2. Developing Advanced Assistive Technologies

- **AI-Powered Assistive Devices:**
  - Using the knowledge acquired from this project, I aim to develop next-generation assistive devices such as **AI-based personal assistants**, **smart glasses**, or **robotic guidance systems** tailored for visually impaired individuals.
  - Enhanced image analysis skills will enable these devices to offer real-time insights, including object recognition, path guidance, and environment interaction.
- **Multi-Modal Systems:**
  - I will integrate audio, visual, and tactile feedback systems into these technologies to.