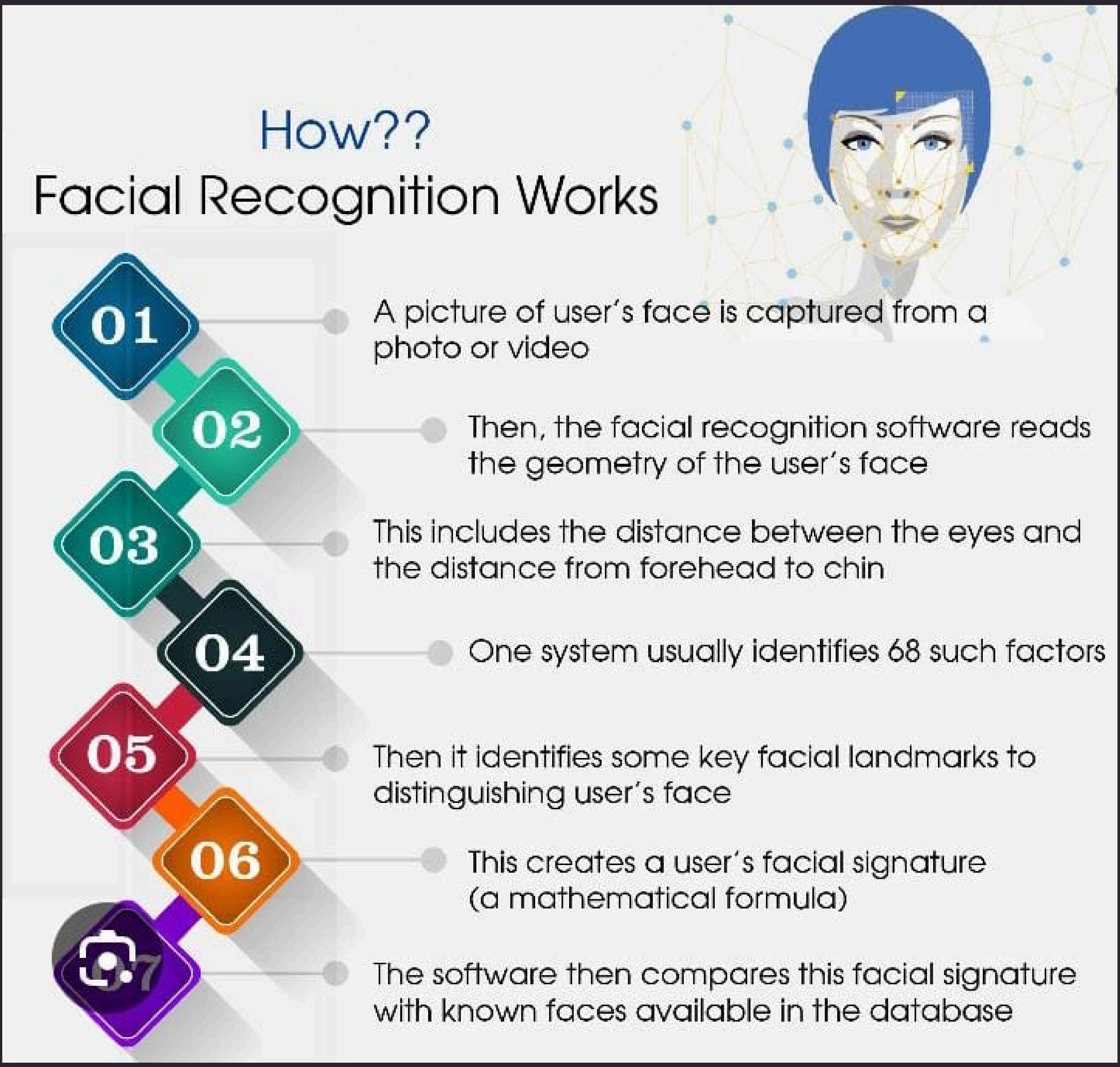
THE UNIVERSITY OF MIRPURKHAS

DO YOU WANT TO KNOW HOW TO

FACE RECOGNITION

HOW FACIAL RECOGNITION WORK



How??
Facial Recognition Works

01 A picture of user's face is captured from a photo or video

02 Then, the facial recognition software reads the geometry of the user's face

03 This includes the distance between the eyes and the distance from forehead to chin

04 One system usually identifies 68 such factors

05 Then it identifies some key facial landmarks to distinguishing user's face

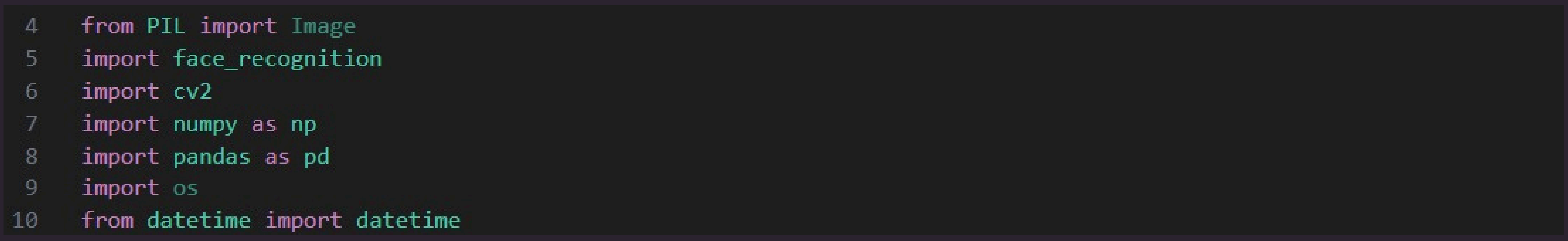
06 This creates a user's facial signature (a mathematical formula)

07 The software then compares this facial signature with known faces available in the database

HOW IT WORK

Facial recognition tech is a system that identifies or verifies individuals by analyzing facial features captured from photos or videos. It reads face geometry, creates a unique facial signature (math formula), and compares it with known faces in a database.

LIBRARIES

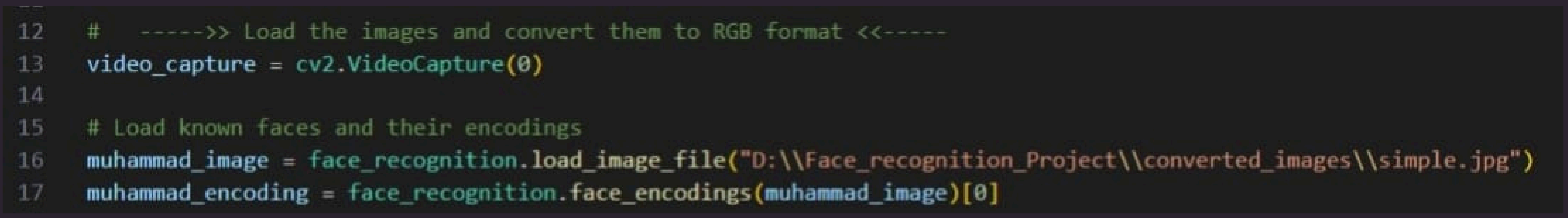


```
from PIL import Image
import face_recognition
import cv2
import numpy as np
import pandas as pd
import os
from datetime import datetime
```

HOW IT WORK

- PIL → for working with images
- face_recognition → for finding and knowing faces
- cv2 → for using the camera
- numpy → for math and image data
- pandas → for saving data
- os → for file handling
- datetime → for date and time

FACE DATA CODE

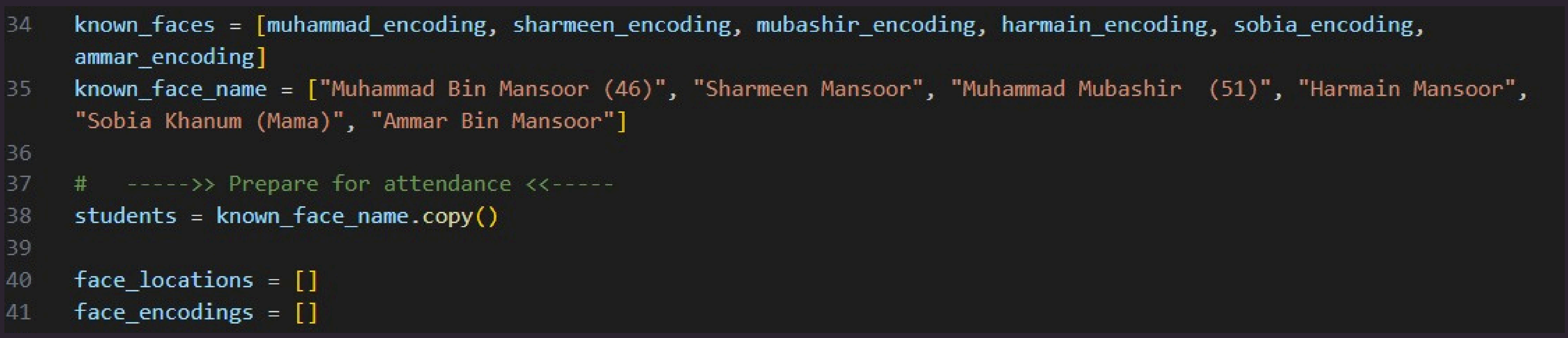


```
#   ----->> Load the images and convert them to RGB format <<-----
video_capture = cv2.VideoCapture(0)

# Load known faces and their encodings
muhammad_image = face_recognition.load_image_file("D:\\Face_recognition_Project\\converted_images\\simple.jpg")
muhammad_encoding = face_recognition.face_encodings(muhammad_image)[0]
```

HOW IT WORK

Turns on the camera
Loads photos of Muhammad, damni, and Mubashir
Saves their face data so the system can recognize them later

FACE LOADED IN LIST

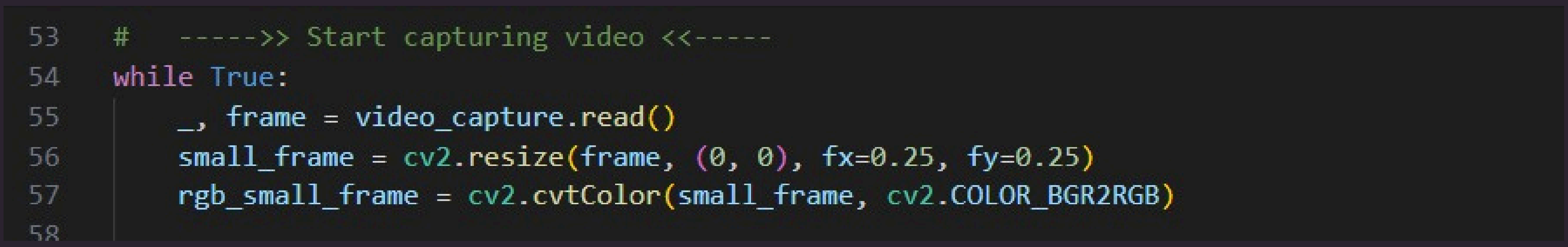


```
known_faces = [muhammad_encoding, sharmeen_encoding, mubashir_encoding, harmain_encoding, sobia_encoding, ammar_encoding]
known_face_name = ["Muhammad Bin Mansoor (46)", "Sharmeen Mansoor", "Muhammad Mubashir  (51)", "Harmain Mansoor", "Sobia Khanum (Mama)", "Ammar Bin Mansoor"]

#   ----->> Prepare for attendance <<-----
students = known_face_name.copy()

face_locations = []
face_encodings = []
```

HOW IT WORK

This code loads an image of Damni from the specified file path, extracts her facial features as a numeric encoding, and stores it for recognition. Then, it creates two lists: one for the known face encodings (used to identify people) and another for their corresponding names or IDs.

FRAME WORK

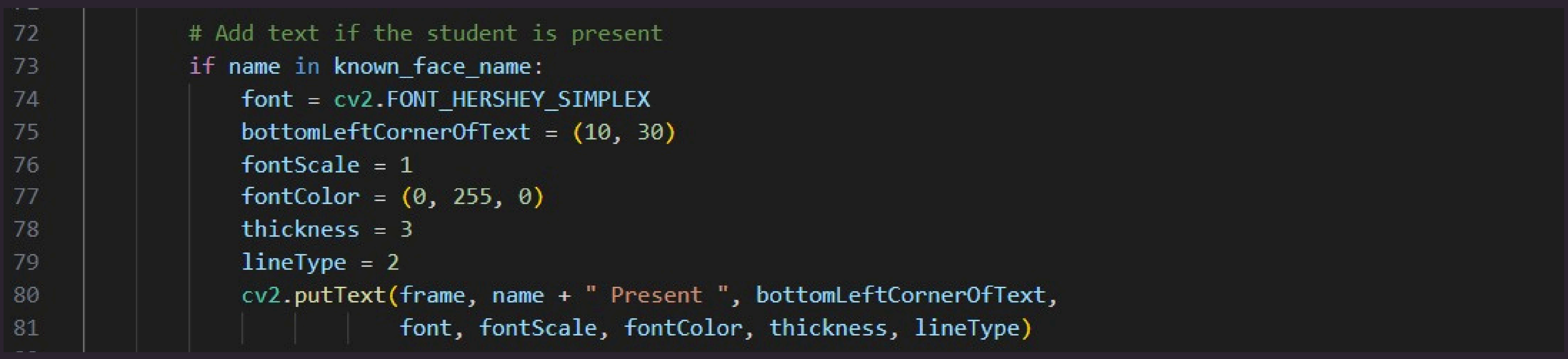


```
#   ----->> Start capturing video <<-----
while True:
    _, frame = video_capture.read()
    small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)
    rgb_small_frame = cv2.cvtColor(small_frame, cv2.COLOR_BGR2RGB)
```

HOW IT WORK

This code captures a single video frame from the webcam, reduces its size to one-quarter for faster processing, and then converts the frame’s color format from BGR (used by OpenCV) to RGB (used by most image-processing libraries).

CAMERA FRAME STRUCTURE

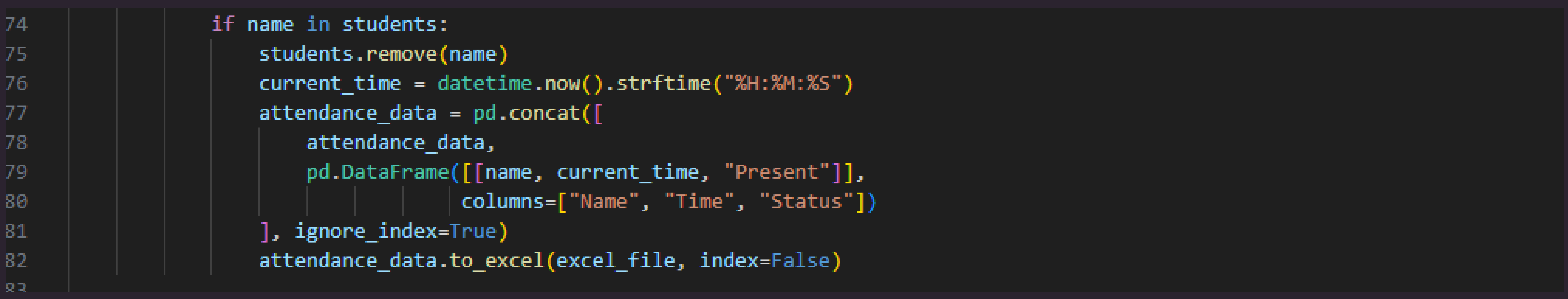


```python
        # Add text if the student is present
        if name in known_face_name:
            font = cv2.FONT_HERSHEY_SIMPLEX
            bottomLeftCornerOfText = (10, 30)
            fontScale = 1
            fontColor = (0, 255, 0)
            thickness = 3
            lineType = 2
            cv2.putText(frame, name + " Present ", bottomLeftCornerOfText,
                        font, fontScale, fontColor, thickness, lineType)
```

HOW IT WORK

This block of code checks if the detected face’s name exists in the list of known names. If it matches, the program displays the person’s name followed by the word “Present” on the video frame. It first sets the font style using cv2.FONT_HERSHEY_SIMPLEX, which provides a simple and readable look. The text position is defined by bottomLeftCornerOfText = (10, 30), meaning the text will appear near the top-left corner of the screen. The size of the text is controlled by fontScale = 1, giving it a medium and balanced appearance. The color (0, 255, 0) represents green, which visually indicates successful detection. The thickness = 3 makes the text bold enough to stand out on the frame, while lineType = 2 ensures smooth, anti-aliased edges. Finally, the cv2.putText() function displays the text on the frame with all these settings applied, showing messages like “Muhammad Present” or “Mubashir Present” whenever their faces are recognized.

EXPORT DATA INTO EXCEL FILE



```python
            if name in students:
                students.remove(name)
                current_time = datetime.now().strftime("%H:%M:%S")
                attendance_data = pd.concat([
                    attendance_data,
                    pd.DataFrame([[name, current_time, "Present"]],
                                 columns=["Name", "Time", "Status"])
                ], ignore_index=True)
                attendance_data.to_excel(excel_file, index=False)
```

HOW IT WORK

This part of the code records a student’s attendance once their face is recognized. It first checks if the detected person’s name exists in the students list. If found, the name is removed to prevent duplicate entries. The datetime.now().strftime("%H:%M:%S") line captures the current time in hours, minutes, and seconds. Then, a new row containing the student’s name, the recorded time, and their status as “Present” is created using a small DataFrame. This new entry is added to the main attendance_data DataFrame using pd.concat(). Finally, the updated attendance sheet is saved to an Excel file using attendance_data.to_excel(excel_file, index=False), ensuring that every recognized student’s attendance is stored with their name, time, and status in the Excel sheet.

THE UNIVERSITY OF MIRPURKHAS

THIS PROJECT DEVELOPED BY (COMPUTER SCIENCE DEPARTMENT) (BATCH-2K25)

This project is a reflection of how students at the University of Mirpurkhas are applying practical programming and AI skills to real world problems. It’s designed for readers who want to understand how face recognition can be used to automate tasks like attendance. Through this work, we aim to show how modern education and teamwork can turn innovative ideas into working solutions that make daily systems smarter and more efficient.

PROJECT PARTICIPANTS

1-MUHAHHAD BIN MANSOOR

2-MUSABBIHA RAJPUT

3-MUHAMMAD MUBASHIR

4-MEMOONA KHAN

5-DAMNI LOHANA

THIS PROJECT CONCLUDES THE SUCCESSFUL IMPLEMENTATION OF A REAL-TIME FACE RECOGNITION ATTENDANCE SYSTEM BY THE CS DEPARTMENT, BATCH 2K25.

THE END