CODE :1

```
import cv2
import os
import numpy as np
from PIL import Image
import datetime
import csv



if not os.path.exists("dataset"):
    os.makedirs("dataset")
if not os.path.exists("trainer"):
    os.makedirs("trainer")
if not os.path.exists("attendance.csv"):
    with open("attendance.csv", "w", newline='') as file:
        writer = csv.writer(file)
        writer.writerow(["Name", "Date", "Time"])



num_people = int(input("How many people's faces do you want to capture? "))


for i in range(num_people):
    while True:
        try:
            student_id = int(input(f"\n[{i+1}/{num_people}] Enter numeric student ID: "))
            break
```

```
    except ValueError:
        print(" ID must be a number!")

name = input("Enter student name: ")
cam = cv2.VideoCapture(0)
detector = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')
count = 0

print(f"\n Capturing 20 images for {name}... Look at the camera.")

while True:
    ret, img = cam.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = detector.detectMultiScale(gray, 1.3, 5)

    for (x, y, w, h) in faces:
        count += 1
        filename = f"dataset/{name}.{student_id}.{count}.jpg"
        cv2.imwrite(filename, gray[y:y+h, x:x+w])
        cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)

    cv2.imshow(f'Capture Face - {name}', img)
    k = cv2.waitKey(100) & 0xff
    if k == 27 or count >= 20:
        break

cam.release()
```

```
    cv2.destroyAllWindows()
    print(f" Saved 20 images for {name}.\n")

print(" Training faces. Please wait...")

recognizer = cv2.face.LBPHFaceRecognizer_create()
detector = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')
id_name_map = {}

def getImagesAndLabels(path):
    imagePaths = [os.path.join(path, f) for f in os.listdir(path) if f.endswith(('jpg', 'jpeg',
'png'))]
    faceSamples = []
    ids = []

    for imagePath in imagePaths:
        try:
            PIL_img = Image.open(imagePath).convert('L')
            img_numpy = np.array(PIL_img, 'uint8')
            parts = os.path.split(imagePath)[-1].split(".")
            name = parts[0]
            id = int(parts[1])
            id_name_map[id] = name
            faces = detector.detectMultiScale(img_numpy)
            for (x, y, w, h) in faces:
                faceSamples.append(img_numpy[y:y+h, x:x+w])
                ids.append(id)
        except Exception as e:
```

```
            print(f" Skipped {imagePath}: {e}")

    return faceSamples, ids

faces, ids = getImagesAndLabels("dataset")
recognizer.train(faces, np.array(ids))
recognizer.write("trainer/trainer.yml")
print("Training complete.\n")



print("📷 Starting real-time face recognition...")

recognizer.read("trainer/trainer.yml")
faceCascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')
font = cv2.FONT_HERSHEY_SIMPLEX
seen_ids = set()

def mark_attendance(id, name):
    if id not in seen_ids:
        now = datetime.datetime.now()
        date = now.strftime("%Y-%m-%d")
        time = now.strftime("%H:%M:%S")
        with open('attendance.csv', 'a', newline='') as file:
            writer = csv.writer(file)
            writer.writerow([name, date, time])
        seen_ids.add(id)
        print(f" Attendance marked for {name}")
```

```
cam = cv2.VideoCapture(0)

while True:
    ret, img = cam.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(gray, 1.2, 5)

    for (x, y, w, h) in faces:
        cv2.rectangle(img, (x, y), (x+w, y+h), (0, 255, 0), 2)
        id, confidence = recognizer.predict(gray[y:y+h, x:x+w])

        if confidence < 80:
            name = id_name_map.get(id, "Unknown")
            mark_attendance(id, name)
        else:
            name = "Unknown"

        confidence_text = f"{round(100 - confidence)}%"
        cv2.putText(img, name, (x+5, y-5), font, 1, (255, 255, 255), 2)
        cv2.putText(img, confidence_text, (x+5, y+h-5), font, 1, (255, 255, 0), 1)

    cv2.imshow("Smart Attendance", img)
    k = cv2.waitKey(10) & 0xff
    if k == 27:
        break

cam.release()
```

```
cv2.destroyAllWindows()

print(" Attendance session ended.")
```

CODE:2

```
import cv2
import numpy as np
from PIL import Image
import os

recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('trainer/trainer.yml')

faceCascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')

id_name_map = {}
dataset_path = "dataset"
for filename in os.listdir(dataset_path):
   if filename.endswith(".jpg"):
      parts = filename.split(".")
      name = parts[0]
      try:
         id = int(parts[1])
         id_name_map[id] = name
      except:
         continue

cam = cv2.VideoCapture(0)
font = cv2.FONT_HERSHEY_SIMPLEX
```

```
print(" Showing recognized names... Press ESC to exit.")

while True:
    ret, img = cam.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(gray, 1.2, 5)

    for (x, y, w, h) in faces:
        id, confidence = recognizer.predict(gray[y:y+h, x:x+w])
        if confidence < 80:
            name = id_name_map.get(id, "Unknown")
        else:
            name = "Unknown"

        confidence_text = f"{round(100 - confidence)}%"
        cv2.rectangle(img, (x, y), (x+w, y+h), (0, 255, 0), 2)
        cv2.putText(img, name, (x+5, y-5), font, 1, (255, 255, 255), 2)
        cv2.putText(img, confidence_text, (x+5, y+h-5), font, 1, (255, 255, 0), 1)

    cv2.imshow("Live Face Recognition", img)

    if cv2.waitKey(10) & 0xFF == 27:
        break

cam.release()
cv2.destroyAllWindows()
print(" Session ended.")
```