```
# -*- coding: utf-8 -*-
"""
Created on Mon Nov  2 08:24:33 2026

@author: sunilsir
"""

import os
import cv2
import face_recognition
import pyttsx3



def speak(text):
  """Freshly initializes pyttsx3 each time to prevent SAPI5 queue lockups on Windows."""
  print(f"[AUDIO] {text}")
  try:
    engine = pyttsx3.init()
    engine.say(text)
    engine.runAndWait()
    engine.stop()
  except Exception as e:
    print(f"[SPEECH ERROR] {e}")
```

```
DATASET_DIR = "dataset"
os.makedirs(DATASET_DIR, exist_ok=True)



def register_face():
 name = input("Enter the name of the person: ").strip()
 if not name:
  print("[ERROR] Name cannot be empty.")
  return

 cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)
 if not cap.isOpened():
  print("[ERROR] Could not open webcam.")
  return

 print(
   "\n[INFO] Look straight at the camera. Press 's' to capture, or 'q' to"
   " cancel."
 )

 while True:
  ret, frame = cap.read()
  if not ret:
   print("[ERROR] Failed to grab frame from camera.")
   break
```

```
cv2.imshow("Register Face - Press 's' to Save", frame)


key = cv2.waitKey(1) & 0xFF
if key == ord("s"):
  rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
  face_locations = face_recognition.face_locations(rgb_frame)


  if len(face_locations) == 0:
    print(
       "[WARNING] No face clearly detected! Please look at the camera and"
       " try again."
    )
    continue


  file_path = os.path.join(DATASET_DIR, f"{name}.jpg")
  cv2.imwrite(file_path, frame)
  print(
     f"[SUCCESS] Face successfully verified and saved for '{name}' at"
     f" {file_path}"
  )
  break
elif key == ord("q"):
  print("[INFO] Registration cancelled.")
  break
```

```
  cap.release()
  cv2.destroyAllWindows()



def recognize_faces():
  known_encodings = []
  known_names = []

  if not os.path.exists(DATASET_DIR) or not os.listdir(DATASET_DIR):
    print(
      "[WARNING] No trained faces found! Please register someone first (Option"
      " 1)."
    )
    return

  print("[INFO] Loading and encoding trained faces...")
  for filename in os.listdir(DATASET_DIR):
    if filename.endswith((".jpg", ".png", ".jpeg")):
      name = os.path.splitext(filename)[0]
      image_path = os.path.join(DATASET_DIR, filename)

      image = face_recognition.load_image_file(image_path)
      encodings = face_recognition.face_encodings(image)
```

```
    if encodings:
      known_encodings.append(encodings[0])
      known_names.append(name)
    else:
      print(f"[WARNING] Could not find a face in {filename}.")

if not known_names:
  print("[ERROR] Loaded 0 valid faces. Please re-register.")
  return

print(f"[SUCCESS] Loaded {len(known_names)} people: {known_names}")

cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)
if not cap.isOpened():
  print("[ERROR] Could not open webcam for recognition.")
  return

print("[INFO] Starting recognition mode. Press 'q' to quit.")
last_spoken_name = None

while True:
  ret, frame = cap.read()
  if not ret:
    print("[WARNING] Can't grab frame. Retrying...")
    continue
```

```
small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)
rgb_small_frame = cv2.cvtColor(small_frame, cv2.COLOR_BGR2RGB)

face_locations = face_recognition.face_locations(rgb_small_frame)
face_encodings = face_recognition.face_encodings(
    rgb_small_frame, face_locations
)

for face_encoding, face_location in zip(face_encodings, face_locations):
  matches = face_recognition.compare_faces(
      known_encodings, face_encoding, tolerance=0.5
  )
  name = "Unknown"

  if True in matches:
    first_match_index = matches.index(True)
    name = known_names[first_match_index]

    if name != last_spoken_name:
      speak(f"Hello {name}")
      last_spoken_name = name

  top, right, bottom, left = [v * 4 for v in face_location]
```

```
      cv2.rectangle(frame, (left, top), (right, bottom), (0, 255, 0), 2)
      cv2.putText(
          frame,
          name,
          (left, top - 10),
          cv2.FONT_HERSHEY_SIMPLEX,
          0.75,
          (0, 255, 0),
          2,
      )

    cv2.imshow("AI Recognition & Greeting (Press 'q' to Quit)", frame)

    if cv2.waitKey(1) & 0xFF == ord("q"):
      break

  cap.release()
  cv2.destroyAllWindows()



if __name__ == "__main__":
  while True:
    print("\n=== AI Desktop Greeter Project ===")
    print("1. Register a new face (Capture from webcam)")
    print("2. Start Recognition & Greeting Mode")
```

```
print("3. Exit")

choice = input("Enter your choice (1/2/3): ").strip()


if choice == "1":

  register_face()

elif choice == "2":

  recognize_faces()

elif choice == "3":

  print("Exiting. Goodbye!")

  break

else:

  print("Invalid choice, please select 1, 2, or 3.")
```