import os

import cv2

import openpyxl

import random

import string

import tkinter as tk

from tkinter import ttk, simpledialog, messagebox

from PIL import Image, ImageTk

*# Path to the folder where face images will be saved*

FACES\_FOLDER = 'captured\_faces'

ATTENDANCE\_FILE = 'attendance.xlsx'

LARGE\_WINDOW\_SIZE = "1000x800"

class FaceImageCaptureApp:

    def \_\_init\_\_(self, root):

*self*.root = root

*self*.root.title('Face Image Capture App')

*self*.root.geometry("400x500")

*self*.student\_id = None

*self*.student\_name = None

*self*.captured\_ids = set()

*self*.image\_id\_dict = {}

*self*.verification\_codes = {}

*# Create input fields for name and course*

*self*.name\_label = ttk.Label(root, text='Name:')

*self*.name\_label.pack()

*self*.name\_entry = ttk.Entry(root,width=60)

*self*.name\_entry.pack()

*self*.course\_label = ttk.Label(root, text='Course:')

*self*.course\_label.pack()

*self*.course\_entry = ttk.Entry(root,width=60)

*self*.course\_entry.pack()

*# Create a button to open the camera frame*

*self*.open\_camera\_button = ttk.Button(root, text='Open Camera', command=*self*.open\_camera\_frame)

*self*.open\_camera\_button.pack(pady=10)

*# Create a button to check attendance*

*self*.check\_attendance\_button = ttk.Button(root, text='Check Attendance', command=*self*.check\_attendance)

*self*.check\_attendance\_button.pack(pady=10)

*# Create a button to restart the app*

*self*.restart\_button = ttk.Button(root, text='Restart', command=*self*.restart\_app)

*self*.restart\_button.pack(pady=10)

    def open\_camera\_frame(self):

*self*.student\_id = *self*.get\_student\_id()

        if not *self*.student\_id:

            return

        if *self*.student\_id in *self*.captured\_ids:

            messagebox.showwarning("Image Captured Before", "Image has been captured before for this student ID.")

            return

        if not os.path.exists(FACES\_FOLDER):

            os.makedirs(FACES\_FOLDER)

*# Check if an image with the same student ID already exists*

        image\_path = f"{FACES\_FOLDER}/{*self*.student\_id}.jpg"

        if os.path.isfile(image\_path):

            messagebox.showwarning("Image Captured Before", "Image has been captured before for this student ID.")

*self*.captured\_ids.add(*self*.student\_id)

            return

*# Generate and display a random verification code*

        verification\_code = ''.join(random.choices(string.ascii\_uppercase + string.digits, k=6))

*self*.verification\_codes[*self*.student\_id] = verification\_code

        messagebox.showinfo("Verification Code", f"Please share this verification code with the person being registered:\n\n{verification\_code}")

*self*.camera = cv2.VideoCapture(0)

*self*.camera\_frame = tk.Toplevel(*self*.root)

*self*.camera\_frame.title("Camera Frame")

*self*.camera\_frame.geometry(LARGE\_WINDOW\_SIZE)

*self*.camera\_label = ttk.Label(*self*.camera\_frame)

*self*.camera\_label.pack()

*self*.capture\_button = ttk.Button(*self*.camera\_frame, text='Capture Face', command=*self*.capture\_face)

*self*.capture\_button.pack(pady=10)

*self*.show\_camera\_frame()

    def show\_camera\_frame(self):

        ret, frame = *self*.camera.read()

        if ret:

            frame = cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB)

            frame = Image.fromarray(frame)

            frame = ImageTk.PhotoImage(frame)

*self*.camera\_label.config(image=frame)

*self*.camera\_label.image = frame

*self*.camera\_frame.after(10, *self*.show\_camera\_frame)

    def capture\_face(self):

        if not *self*.student\_id:

            messagebox.showwarning("Student ID Not Set", "Please enter the student ID and click 'Open Camera' first.")

            return

        ret, frame = *self*.camera.read()

        if not ret:

            messagebox.showerror("Error", "Unable to access the camera.")

            return

*# Detect face in the frame*

        face\_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade\_frontalface\_default.xml')

        gray\_frame = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

        faces = face\_cascade.detectMultiScale(gray\_frame, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))

        if len(faces) > 0:

*# Save the face image to the folder*

            for (x, y, w, h) in faces:

                face\_image = gray\_frame[y:y+h, x:x+w]

                face\_image\_path = f"{FACES\_FOLDER}/{*self*.student\_id}.jpg"

*# Check if the image is already associated with another student ID*

                if face\_image\_path in *self*.image\_id\_dict:

                    existing\_student\_id = *self*.image\_id\_dict[face\_image\_path]

                    messagebox.showwarning("Image Already Used",

                                            f"This image is already associated with Student ID: {existing\_student\_id}.")

                    return

                cv2.imwrite(face\_image\_path, face\_image)

                messagebox.showinfo("Image Captured", f"Face image for Student ID: {*self*.student\_id} captured successfully.")

*self*.captured\_ids.add(*self*.student\_id)  *# Add the student ID to the set*

*self*.image\_id\_dict[face\_image\_path] = *self*.student\_id

*self*.save\_attendance\_info()

                break

        else:

            messagebox.showwarning("No Face Detected", "No face detected in the captured image.")

    def save\_attendance\_info(self):

        name = *self*.name\_entry.get()

        course = *self*.course\_entry.get()

        if not name or not course:

            messagebox.showwarning("Incomplete Information", "Please enter both name and course.")

            return

*# Check verification code before saving attendance info*

        if *self*.student\_id in *self*.verification\_codes:

            verification\_code = simpledialog.askstring("Verification Code", "Please enter the verification code:")  *# Adjust font size as needed)*

            if verification\_code is None or verification\_code != *self*.verification\_codes[*self*.student\_id]:

                messagebox.showwarning("Invalid Verification Code", "Invalid verification code. Registration is not confirmed.")

                return

            del *self*.verification\_codes[*self*.student\_id]  *# Remove the verification code entry after successful registration*

        workbook = openpyxl.load\_workbook(ATTENDANCE\_FILE)

        sheet = workbook.active

        row = (name, course, *self*.student\_id)

        sheet.append(row)

        workbook.save(ATTENDANCE\_FILE)

    def check\_attendance(self):

        if not *self*.student\_id:

            messagebox.showwarning("Student ID Not Set", "Please enter the student ID and click 'Open Camera' first.")

            return

        image\_path = f"{FACES\_FOLDER}/{*self*.student\_id}.jpg"

        if os.path.isfile(image\_path):

*# Load attendance details from the file*

            workbook = openpyxl.load\_workbook(ATTENDANCE\_FILE)

            sheet = workbook.active

*# Find the row with the student ID*

            for row in sheet.iter\_rows(values\_only=True):

                if row[2] == *self*.student\_id:

*self*.student\_name = row[0]

                    break

            messagebox.showinfo("Attendance Information", f"Student ID: {*self*.student\_id}\nName: {*self*.student\_name}")

        else:

            messagebox.showwarning("Image Not Found", "No image found for the provided student ID.")

    def get\_student\_id(self):

        student\_id = simpledialog.askstring("Student ID", "Please enter the student ID:")  *# Adjust font size as needed)*

        if student\_id is None:

            return None

        if not student\_id.strip():

            messagebox.showwarning("Invalid Input", "Please enter a valid student ID.")

            return None

        return student\_id.strip()

    def restart\_app(self):

*self*.root.destroy()

        root = tk.Tk()

        app = FaceImageCaptureApp(root)

        root.mainloop()

    def \_\_del\_\_(self):

        if hasattr(*self*, 'camera'):

*self*.camera.release()

if \_\_name\_\_ == "\_\_main\_\_":

    if not os.path.exists(FACES\_FOLDER):

        os.makedirs(FACES\_FOLDER)

    if not os.path.exists(ATTENDANCE\_FILE):

        workbook = openpyxl.Workbook()

        sheet = workbook.active

        sheet.append(['Name', 'Course', 'Student ID'])

        workbook.save(ATTENDANCE\_FILE)

    root = tk.Tk()

    app = FaceImageCaptureApp(root)

    root.mainloop()