import tkinter as tk

from tkinter import messagebox, ttk,simpledialog

from PIL import Image, ImageTk

import mysql.connector

import random

import os

DB\_HOST = "localhost"

DB\_USER = "root"

DB\_PASSWORD = ""

DB\_NAME = "vault"

def connect\_db():

return mysql.connector.connect(host=DB\_HOST, user=DB\_USER, password=DB\_PASSWORD, database=DB\_NAME)

def generate\_health\_id(title, username):

unique\_number = random.randint(100000, 999999)

if title == "Dr.":

return f"DOC{unique\_number}"

else:

initials = "".join([x[0].upper() for x in username.split() if x])

return f"{initials}{unique\_number}"

def signup\_user():

title = title\_var.get()

username = username\_entry.get().strip()

email = email\_entry.get().strip()

identification\_id = id\_entry.get().strip()

dob = dob\_entry.get().strip()

password = password\_entry.get().strip()

if not all([username, email, identification\_id, dob, password]):

messagebox.showerror("Error", "Please fill all fields!")

return

if "@" not in email:

messagebox.showerror("Error", "Invalid Email Format!")

return

health\_id = generate\_health\_id(title, username)

try:

conn = connect\_db()

cursor = conn.cursor()

if title == "Dr.":

cursor.execute("INSERT INTO doctors (health\_id, title, username, email, identification\_id, dob, password) VALUES (%s, %s, %s, %s, %s, %s, %s)",

(health\_id, title, username, email, identification\_id, dob, password))

else:

cursor.execute("INSERT INTO patients (health\_id, title, username, email, identification\_id, dob, password) VALUES (%s, %s, %s, %s, %s, %s, %s)",

(health\_id, title, username, email, identification\_id, dob, password))

conn.commit()

messagebox.showinfo("Success", f"Signup Successful! Your Health ID: {health\_id}")

except mysql.connector.Error as e:

messagebox.showerror("Error", f"Database Error: {e}")

finally:

conn.close()

def open\_login\_page():

root.withdraw()

login\_window.deiconify()

def view\_personal\_info():

password = simpledialog.askstring("Authentication", "Enter your password:", show="\*")

if not password:

messagebox.showerror("Error", "Password is required!")

return

conn = connect\_db()

cursor = conn.cursor()

cursor.execute("SELECT title, username, health\_id, dob, email FROM doctors WHERE password = %s", (password,))

doctor = cursor.fetchone()

if doctor:

info = f"Name: {doctor[0]} {doctor[1]}\nHealth ID: {doctor[2]}\nDate of Birth: {doctor[3]}\nEmail: {doctor[4]}"

messagebox.showinfo("Personal Information", info)

conn.close()

return

cursor.execute("SELECT title, username, health\_id, dob, email FROM patients WHERE password = %s", (password,))

patient = cursor.fetchone()

if patient:

info = f"Name: {patient[0]} {patient[1]}\nHealth ID: {patient[2]}\nDate of Birth: {patient[3]}\nEmail: {patient[4]}"

messagebox.showinfo("Personal Information", info)

else:

messagebox.showerror("Error", "Invalid password!")

conn.close()

def open\_patient\_dashboard():

login\_window.withdraw()

dashboard = tk.Toplevel()

dashboard.title("General Dashboard")

dashboard.geometry("450x350")

dashboard.configure(bg="#F4F4F4")

header = tk.Frame(dashboard, bg="#2E86C1", height=50)

header.pack(fill="x")

tk.Label(header, text="Patient Dashboard", font=("Arial", 16, "bold"), bg="#2E86C1", fg="white").pack(pady=10)

content\_frame = tk.Frame(dashboard, bg="#F4F4F4")

content\_frame.pack(pady=20)

btn\_style = {"font": ("Arial", 12, "bold"), "width": 25, "height": 2, "bd": 2, "relief": "raised"}

tk.Button(content\_frame, text="📄 Personal Info", \*\*btn\_style, bg="#3498DB", fg="white",

command=view\_personal\_info).pack(pady=8)

tk.Button(content\_frame, text="💊 Prescriptions", \*\*btn\_style, bg="#2ECC71", fg="white",

command=lambda: messagebox.showinfo("Info", "Prescriptions Page")).pack(pady=8)

tk.Button(content\_frame, text="📂 Medical Reports", \*\*btn\_style, bg="#E74C3C", fg="white",

command=lambda: messagebox.showinfo("Info", "Medical Reports Page")).pack(pady=8)

tk.Button(content\_frame, text="🗓 Appointment", \*\*btn\_style, bg="#E74C3C", fg="white",

command=lambda: messagebox.showinfo("Info", "Medical Reports Page")).pack(pady=8)

footer = tk.Frame(dashboard, bg="#D5DBDB", height=30)

footer.pack(side="bottom", fill="x")

tk.Label(footer, text=" AI-Based EHR System", bg="#D5DBDB", font=("Arial", 10)).pack(pady=5)

dashboard.mainloop()

def open\_doctor\_dashboard():

login\_window.withdraw()

doctor\_dashboard = tk.Toplevel()

doctor\_dashboard.title("Doctor Dashboard")

doctor\_dashboard.geometry("450x350")

doctor\_dashboard.configure(bg="#F4F4F4")

header = tk.Frame(doctor\_dashboard, bg="#2E86C1", height=50)

header.pack(fill="x")

tk.Label(header, text="Doctor Dashboard", font=("Arial", 16, "bold"), bg="#2E86C1", fg="white").pack(pady=10)

content\_frame = tk.Frame(doctor\_dashboard, bg="#F4F4F4")

content\_frame.pack(pady=20)

btn\_style = {"font": ("Arial", 12, "bold"), "width": 25, "height": 2, "bd": 2, "relief": "raised"}

tk.Button(content\_frame, text="📄 Personal Info", \*\*btn\_style, bg="#3498DB", fg="white", command=view\_personal\_info).pack(pady=8)

tk.Button(content\_frame, text="🩺 Patients", \*\*btn\_style, bg="#2ECC71", fg="white", command=open\_patient\_management).pack(pady=8)

tk.Button(content\_frame, text="✍ Make Prescriptions", \*\*btn\_style, bg="#E74C3C", fg="white").pack(pady=8)

tk.Button(content\_frame, text="📂 Make/View Reports", \*\*btn\_style, bg="#F39C12", fg="white").pack(pady=8)

footer = tk.Frame(doctor\_dashboard, bg="#D5DBDB", height=30)

footer.pack(side="bottom", fill="x")

tk.Label(footer, text="AI-Based EHR System", bg="#D5DBDB", font=("Arial", 10)).pack(pady=5)

doctor\_dashboard.mainloop()

def add\_patient():

def save\_patient():

patient\_name = patient\_name\_entry.get().strip()

health\_id = health\_id\_entry.get().strip()

doctor\_id = doctor\_id\_entry.get().strip()

if not patient\_name or not health\_id or not doctor\_id:

messagebox.showerror("Error", "All fields are required!")

return

try:

conn = connect\_db()

cursor = conn.cursor()

cursor.execute("SELECT health\_id FROM patients WHERE username=%s AND health\_id=%s", (patient\_name, health\_id))

result = cursor.fetchone()

if result:

cursor.execute("INSERT INTO treatment (patient, doctor) VALUES (%s, %s)", (health\_id, doctor\_id))

conn.commit()

messagebox.showinfo("Success", "Patient added to treatment successfully!")

add\_window.destroy()

else:

messagebox.showerror("Error", "Patient not found in records!")

except mysql.connector.Error as e:

messagebox.showerror("Error", f"Database Error: {e}")

finally:

conn.close()

add\_window = tk.Toplevel()

add\_window.title("Add Patient")

add\_window.geometry("350x250")

add\_window.configure(bg="#F4F4F4")

tk.Label(add\_window, text="Patient Name:", bg="#F4F4F4").pack(pady=5)

patient\_name\_entry = tk.Entry(add\_window)

patient\_name\_entry.pack(pady=5)

tk.Label(add\_window, text="Health ID:", bg="#F4F4F4").pack(pady=5)

health\_id\_entry = tk.Entry(add\_window)

health\_id\_entry.pack(pady=5)

tk.Label(add\_window, text="Doctor ID:", bg="#F4F4F4").pack(pady=5)

doctor\_id\_entry = tk.Entry(add\_window)

doctor\_id\_entry.pack(pady=5)

submit\_button = tk.Button(add\_window, text="Submit", command=save\_patient, bg="#3498DB", fg="white")

submit\_button.pack(pady=10)

def view\_patients():

def fetch\_patients():

doctor\_id = doctor\_id\_entry.get().strip()

if not doctor\_id:

messagebox.showerror("Error", "Please enter a Doctor ID!")

return

try:

conn = connect\_db()

cursor = conn.cursor()

cursor.execute("""

SELECT patients.username

FROM patients

JOIN treatment ON patients.health\_id = treatment.patient

WHERE treatment.doctor = %s

""", (doctor\_id,))

patients = cursor.fetchall()

conn.close()

if patients:

result\_text.set("\n".join([p[0] for p in patients]))

else:

result\_text.set("No patients found for this doctor.")

except mysql.connector.Error as e:

messagebox.showerror("Error", f"Database Error: {e}")

view\_window = tk.Toplevel()

view\_window.title("View Patients")

view\_window.geometry("350x250")

view\_window.configure(bg="#F4F4F4")

tk.Label(view\_window, text="Enter Doctor ID:", bg="#F4F4F4").pack(pady=5)

doctor\_id\_entry = tk.Entry(view\_window)

doctor\_id\_entry.pack(pady=5)

search\_button = tk.Button(view\_window, text="View Patients", command=fetch\_patients, bg="#27AE60", fg="white")

search\_button.pack(pady=10)

result\_text = tk.StringVar()

result\_label = tk.Label(view\_window, textvariable=result\_text, bg="#F4F4F4", fg="black")

result\_label.pack(pady=10)

def open\_patient\_management():

patient\_window = tk.Toplevel()

patient\_window.title("Manage Patients")

patient\_window.geometry("400x250")

patient\_window.configure(bg="#F4F4F4")

tk.Label(patient\_window, text="Patient Management", font=("Arial", 14, "bold"), bg="#F4F4F4").pack(pady=10)

tk.Button(patient\_window, text="➕ Add Patient", font=("Arial", 12), width=20, command=add\_patient).pack(pady=10)

tk.Button(patient\_window, text="📋 View Patients", font=("Arial", 12), width=20, command=view\_patients).pack(pady=10)

patient\_window.mainloop()

def login\_user():

health\_id = login\_health\_id\_entry.get().strip()

password = login\_password\_entry.get().strip()

if not health\_id or not password:

messagebox.showerror("Error", "Please enter Health ID and Password!")

return

try:

conn = connect\_db()

cursor = conn.cursor()

cursor.execute("SELECT username FROM doctors WHERE health\_id = %s AND password = %s", (health\_id, password))

doctor = cursor.fetchone()

cursor.execute("SELECT username FROM patients WHERE health\_id = %s AND password = %s", (health\_id, password))

patient = cursor.fetchone()

if doctor:

messagebox.showinfo("Success", f"Welcome Dr. {doctor[0]}! Login Successful!")

open\_doctor\_dashboard()

elif patient:

messagebox.showinfo("Success", f"Welcome {patient[0]}! Login Successful!")

open\_patient\_dashboard()

else:

messagebox.showerror("Error", "Invalid Health ID or Password!")

except mysql.connector.Error as e:

messagebox.showerror("Error", f"Database Error: {e}")

finally:

conn.close()

root = tk.Tk()

root.title("Hospital Signup System")

root.geometry("600x600")

def load\_bg\_image(image\_path):

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

img = Image.open(image\_path)

img = img.resize((600, 600), Image.LANCZOS)

return ImageTk.PhotoImage(img)

else:

messagebox.showerror("Error", "Background image not found!")

return None

bg\_image\_path = r"C:\Users\User\Desktop\AI\_EHR\hospital\_bg.jpg"

bg\_photo = load\_bg\_image(bg\_image\_path)

if bg\_photo:

bg\_label = tk.Label(root, image=bg\_photo)

bg\_label.place(x=0, y=0, relwidth=1, relheight=1)

title\_var = tk.StringVar(value="Mr.")

tk.Label(root, text="Title:").place(x=50, y=50)

title\_dropdown = ttk.Combobox(root, textvariable=title\_var, values=["Mr.", "Mrs.", "Dr."])

title\_dropdown.place(x=200, y=50)

tk.Label(root, text="Username:").place(x=50, y=90)

username\_entry = tk.Entry(root)

username\_entry.place(x=200, y=90)

tk.Label(root, text="Email:").place(x=50, y=130)

email\_entry = tk.Entry(root)

email\_entry.place(x=200, y=130)

tk.Label(root, text="ID:").place(x=50, y=170)

id\_entry = tk.Entry(root)

id\_entry.place(x=200, y=170)

tk.Label(root, text="DOB (YYYY-MM-DD):").place(x=50, y=210)

dob\_entry = tk.Entry(root)

dob\_entry.place(x=200, y=210)

tk.Label(root, text="Password:").place(x=50, y=250)

password\_entry = tk.Entry(root, show="\*")

password\_entry.place(x=200, y=250)

signup\_btn = tk.Button(root, text="Sign Up", command=signup\_user)

signup\_btn.place(x=150, y=350)

login\_btn = tk.Button(root, text="Login", command=open\_login\_page)

login\_btn.place(x=250, y=350)

login\_window = tk.Toplevel(root)

login\_window.title("Login")

login\_window.geometry("400x300")

login\_window.withdraw()

tk.Label(login\_window, text="Health ID:").place(x=50, y=50)

login\_health\_id\_entry = tk.Entry(login\_window)

login\_health\_id\_entry.place(x=180, y=50)

tk.Label(login\_window, text="Password:").place(x=50, y=100)

login\_password\_entry = tk.Entry(login\_window, show="\*")

login\_password\_entry.place(x=180, y=100)

login\_submit\_btn = tk.Button(login\_window, text="Login", command=login\_user)

login\_submit\_btn.place(x=150, y=150)

root.mainloop()