CODE OF PROJECT

import tkinter as tk

from tkinter import ttk

from tkinter import messagebox

import tkinter as Tk

import os

import sys

# Configure style

def configure\_style():

    style = ttk.Style()

    style.configure('Custom.TEntry', padding=10)

    style.configure('Custom.TCombobox', padding=10)

    return style

# Create custom button style

class CustomButton(tk.Button):

    def \_\_init\_\_(self, master=None, \*\*kwargs):

        super().\_\_init\_\_(master, \*\*kwargs)

        self.configure(

            bg='#FF9900',  # Amazon's orange color

            fg='white',

            activebackground='#FF8C00',

            activeforeground='white',

            relief='flat',

            cursor='hand2'

        )

        self.bind('<Enter>', lambda e: self.configure(bg='#FF8C00'))

        self.bind('<Leave>', lambda e: self.configure(bg='#FF9900'))

sys.path.insert(0, os.path.join(os.path.dirname(\_\_file\_\_), 'windows'))

import timetable\_stud

import timetable\_fac

import sqlite3

def challenge():

    db\_path = os.path.join(os.path.dirname(\_\_file\_\_), 'files', 'timetable.db')

    conn = sqlite3.connect(db\_path)

    try:

        # First drop and recreate tables to ensure clean state

        conn.execute("DROP TABLE IF EXISTS SCHEDULE")

        conn.execute("DROP TABLE IF EXISTS SUBJECTS")

        conn.execute("DROP TABLE IF EXISTS STUDENT")

        conn.execute("DROP TABLE IF EXISTS FACULTY")

        # Create tables

        conn.execute("""

            CREATE TABLE SUBJECTS (

                SUBCODE TEXT PRIMARY KEY,

                SUBNAME TEXT,

                SUBTYPE TEXT

            )

        """)

        conn.execute("""

            CREATE TABLE FACULTY (

                FID TEXT PRIMARY KEY,

                PASSW TEXT,

                INI TEXT,

                NAME TEXT,

                EMAIL TEXT

            )

        """)

        conn.execute("""

            CREATE TABLE STUDENT (

                SID TEXT PRIMARY KEY,

                PASSW TEXT,

                NAME TEXT,

                SECTION TEXT,

                ROLL TEXT

            )

        """)

        conn.execute("""

            CREATE TABLE SCHEDULE (

                ID TEXT,

                DAYID INTEGER,

                PERIODID INTEGER,

                SECTION TEXT,

                SUBCODE TEXT,

                FINI TEXT

            )

        """)

        # Add subjects

        conn.execute("""

            INSERT INTO SUBJECTS (SUBCODE, SUBNAME, SUBTYPE)

            VALUES

            ('CS101', 'Control Systems', 'T'),

            ('AN101', 'Antennas', 'T'),

            ('ENT101', 'ENT', 'P')

        """)

        # Add faculty

        conn.execute("""

            INSERT INTO FACULTY (FID, PASSW, INI, NAME, EMAIL)

            VALUES ('DIWAKARAN', '1234', 'CSF', 'Control Systems Faculty', 'cs@example.com')

        """)

        # Add student

        conn.execute("""

            INSERT INTO STUDENT (SID, PASSW, NAME, SECTION, ROLL)

            VALUES ('Likhitha', '12345', 'Likhitha', 'A', '1')

        """)

        # Add schedule for all days

        schedule\_data = [

            # Monday

            (0, 0, 'A', 'CS101', 'CSF'), (0, 1, 'A', 'CS101', 'CSF'),

            (0, 2, 'A', 'AN101', 'CSF'), (0, 3, 'A', 'AN101', 'CSF'),

            (0, 4, 'A', 'ENT101', 'CSF'),

            # Tuesday

            (1, 2, 'A', 'CS101', 'CSF'), (1, 3, 'A', 'CS101', 'CSF'),

            (1, 0, 'A', 'AN101', 'CSF'), (1, 1, 'A', 'AN101', 'CSF'),

            (1, 5, 'A', 'ENT101', 'CSF'),

            # Wednesday

            (2, 1, 'A', 'CS101', 'CSF'), (2, 2, 'A', 'CS101', 'CSF'),

            (2, 4, 'A', 'AN101', 'CSF'), (2, 5, 'A', 'AN101', 'CSF'),

            (2, 0, 'A', 'ENT101', 'CSF'),

            # Thursday

            (3, 3, 'A', 'CS101', 'CSF'), (3, 4, 'A', 'CS101', 'CSF'),

            (3, 0, 'A', 'AN101', 'CSF'), (3, 1, 'A', 'AN101', 'CSF'),

            (3, 2, 'A', 'ENT101', 'CSF'),

            # Friday

            (4, 0, 'A', 'CS101', 'CSF'), (4, 1, 'A', 'CS101', 'CSF'),

            (4, 2, 'A', 'AN101', 'CSF'), (4, 3, 'A', 'AN101', 'CSF'),

            (4, 5, 'A', 'ENT101', 'CSF')

        ]

        for day, period, section, subcode, fini in schedule\_data:

            conn.execute("""

                INSERT INTO SCHEDULE (DAYID, PERIODID, SECTION, SUBCODE, FINI)

                VALUES (?, ?, ?, ?, ?)

            """, (day, period, section, subcode, fini))

        conn.commit()

    except Exception as e:

        print(f"Error: {e}")

        conn.rollback()

    user = str(combo1.get())

    if user == "Student":

        cursor = conn.execute(f"SELECT PASSW, SECTION, NAME, ROLL FROM STUDENT WHERE SID='{id\_entry.get()}'")

        cursor = list(cursor)

        if len(cursor) == 0:

            messagebox.showwarning('Bad id', 'No such user found!')

        elif passw\_entry.get() != cursor[0][0]:

            messagebox.showerror('Bad pass', 'Incorret Password!')

        else:

            nw = tk.Tk()

            tk.Label(

                nw,

                text=f'{cursor[0][2]}\tSection: {cursor[0][1]}\tRoll No.: {cursor[0][3]}',

                font=('Consolas', 12, 'italic'),

            ).pack()

            m.destroy()

            timetable\_stud.student\_tt\_frame(nw, cursor[0][1])

            nw.mainloop()

    elif user == "Faculty":

        cursor = conn.execute(f"SELECT PASSW, INI, NAME, EMAIL FROM FACULTY WHERE FID='{id\_entry.get()}'")

        cursor = list(cursor)

        if len(cursor) == 0:

            messagebox.showwarning('Bad id', 'No such user found!')

        elif passw\_entry.get() != cursor[0][0]:

            messagebox.showerror('Bad pass', 'Incorret Password!')

        else:

            nw = tk.Tk()

            tk.Label(

                nw,

                text=f'{cursor[0][2]} ({cursor[0][1]})\tEmail: {cursor[0][3]}',

                font=('Consolas', 12, 'italic'),

            ).pack()

            m.destroy()

            timetable\_fac.fac\_tt\_frame(nw, cursor[0][1])

            nw.mainloop()

    elif user == "Admin":

        if id\_entry.get() == 'jessey' and passw\_entry.get() == '143':

            m.destroy()

            admin\_path = os.path.join(os.path.dirname(\_\_file\_\_), 'windows', 'admin\_screen.py')

            os.system(f'python "{admin\_path}"')

        else:

            messagebox.showerror('Bad Input', 'Incorret Username/Password!')

m = tk.Tk()

m.geometry('500x700')  # More compact size

m.title('Timetable Management System - Login')

m.configure(bg='#FFFFFF')  # White background

# Create main frame with padding

main\_frame = tk.Frame(m, bg='#FFFFFF', padx=40, pady=20)

main\_frame.pack(expand=True, fill='both')

# Footer (Move this before other elements)

footer\_label = tk.Label(

    main\_frame,

    text='DEVELOPED BY GAJULLAPALLI LIKHITHA REDDY & TEAM',

    font=('Arial', 15, 'bold'),  # Increased font size for better visibility

    fg='#444444',  # Darker color for better contrast

    bg='#FFFFFF'

)

footer\_label.pack(side='bottom', pady=30)

# Logo and Title

title\_frame = tk.Frame(main\_frame, bg='#FFFFFF')

title\_frame.pack(pady=20)

tk.Label(

    title\_frame,

    text='TMS',

    font=('Arial Black', 40, 'bold'),

    fg='#FF9900',  # Amazon orange

    bg='#FFFFFF'

).pack()

tk.Label(

    title\_frame,

    text='Timetable Management System',

    font=('Arial', 12),

    fg='#444444',

    bg='#FFFFFF'

).pack()

# Login frame

login\_frame = tk.Frame(main\_frame, bg='#FFFFFF')

login\_frame.pack(pady=20)

# Username

tk.Label(

    login\_frame,

    text='Username',

    font=('Arial', 12, 'bold'),

    fg='#444444',

    bg='#FFFFFF',

    anchor='w'

).pack(fill='x')

id\_entry = tk.Entry(

    login\_frame,

    font=('Arial', 12),

    bd=1,

    relief='solid',

    bg='#FAFAFA'

)

id\_entry.pack(fill='x', pady=(5, 15))

id\_entry.configure(highlightthickness=1, highlightcolor='#FF9900')

# Password

tk.Label(

    login\_frame,

    text='Password',

    font=('Arial', 12, 'bold'),

    fg='#444444',

    bg='#FFFFFF',

    anchor='w'

).pack(fill='x')

pass\_frame = tk.Frame(login\_frame, bg='#FFFFFF')

pass\_frame.pack(fill='x', pady=(5, 15))

passw\_entry = tk.Entry(

    pass\_frame,

    font=('Arial', 12),

    show="●",

    bd=1,

    relief='solid',

    bg='#FAFAFA'

)

passw\_entry.pack(side='left', expand=True, fill='x')

passw\_entry.configure(highlightthickness=1, highlightcolor='#FF9900')

# Show/Hide password button

def show\_passw():

    if passw\_entry['show'] == "●":

        passw\_entry['show'] = ""

        B1\_show.configure(text='Hide')

    else:

        passw\_entry['show'] = "●"

        B1\_show.configure(text='Show')

B1\_show = tk.Button(

    pass\_frame,

    text='Show',

    font=('Arial', 10),

    command=show\_passw,

    bd=0,

    bg='#FFFFFF',

    fg='#0066c0',

    activeforeground='#c45500',

    cursor='hand2'

)

B1\_show.pack(side='left', padx=5)

# User Type Selection

tk.Label(

    login\_frame,

    text='Login as',

    font=('Arial', 12, 'bold'),

    fg='#444444',

    bg='#FFFFFF',

    anchor='w'

).pack(fill='x')

combo1 = ttk.Combobox(

    login\_frame,

    values=['Student', 'Faculty', 'Admin'],

    font=('Arial', 12),

    state='readonly'

)

combo1.pack(fill='x', pady=(5, 20))

combo1.current(0)

# Login button

CustomButton(

    login\_frame,

    text='Sign In',

    font=('Arial', 14, 'bold'),

    command=challenge,

    width=20,

    pady=10

).pack(pady=20)

# Divider

tk.Frame(

    login\_frame,

    height=1,

    bg='#DDDDDD'

).pack(fill='x', pady=20)

m.mainloop()