```python
#-------------------------------------------------------IMPORT LIBRARIES-------------------------------------------------------------------

from tkinter import *
import tkinter.font as font
from PIL import Image, ImageTk
import sqlite3
from tkinter import messagebox
import serial
import bluetooth
import sys
import os

#for converting to windows software
def resource_path(relative_path):
    """ Get absolute path to resource, works for dev and for PyInstaller """
    try:
        # PyInstaller creates a temp folder and stores path in _MEIPASS
        base_path = sys._MEIPASS2
    except Exception:
        base_path = os.path.abspath(".")

    return os.path.join(base_path, relative_path)


#---------------------------------------------------REGISTRATION SECTION--------------------------------------------------------------------

#saves the registered values into the database
def register_user():
    name_info=name.get()
    age_info=age.get()
    conn_type_of_devices_info=conn_type_of_devices.get()
    conn_type_of_devices_info_1=conn_type_of_devices_1.get()
    gender_info=gender.get()
    username_info=username.get()
    password_info= password.get()

    #software.db contains registered info
    conn= sqlite3.connect(resource_path("software.db"))
    c=conn.cursor()
    c.execute("INSERT INTO person VALUES('"+name_info+"','"+age_info+"','"+conn_type_of_devices_info+"','"+conn_type_of_devices_info_1+"','"+gender_info+"','"+username_info+"','"+password_info+"')")
    messagebox.showinfo("Information","Your record is saved!")
    conn.commit()
    conn.close()
    conn= sqlite3.connect(resource_path("user_info.db"))
    c=conn.cursor()
    c.execute("INSERT INTO Curr_session(name) VALUES('"+name_info+"')")
    conn.commit()
    conn.close()

    name_entry.delete(0,END)
    age_entry.delete(0,END)
    conn_devices_entry.deselect()
    conn_devices_entry1.deselect()
    gen_entry_m.deselect()
    gen_entry_f.select()
    username_entry.delete(0,END)
    password_entry.delete(0,END)

    Label(screen1,text="Registeration is Successful", fg="green", font=("Calibri",11)).pack()


#these functions delete the screen
def delete_s1():
    screen1.destroy()

#for the registration page
def register():
    global screen1
    screen1= Toplevel(screen)
    screen1.title("Register")
    screen1.geometry("400x650")
    screen1.configure(bg="#FFFEF2")

    global name
    global age
    global conn_type_of_devices
    global conn_type_of_devices_1
    global gender
    global username
    global password

    global name_entry
    global age_entry
    global conn_devices_entry
    global conn_devices_entry1
    global gen_entry_m
    global gen_entry_f
    global username_entry
    global password_entry

    name=StringVar()
    age=StringVar()
    conn_type_of_devices=StringVar()
    conn_type_of_devices_1=StringVar()
    gender=StringVar()
    username= StringVar()
    password= StringVar()

    Label(screen1, text="Please Enter details below ",width="400",height="3",bg="#caf0f8", font=("Times",16)).pack()

    Label(screen1, text="",bg="#FFFEF2").pack()

    Label(screen1, text="Enter Name",font=("Times",13),bg="#FFFEF2").pack()

    name_entry= Entry(screen1,textvariable=name, width=30)
    name_entry.pack()

    Label(screen1, text="Enter Age",font=("Times",13),bg="#FFFEF2").pack()

    age_entry= Entry(screen1,textvariable=age, width=30)
    age_entry.pack()

    Label(screen1, text="Select types of connection",font=("Times",13),bg="#FFFEF2").pack()

    conn_devices_entry= Checkbutton(screen1, text="Wifi",variable=conn_type_of_devices,bg="#FFFEF2")
    conn_devices_entry.deselect()
    conn_devices_entry.pack()

    conn_devices_entry1= Checkbutton(screen1, text="Bluetooth",variable=conn_type_of_devices_1,bg="#FFFEF2")
    conn_devices_entry1.deselect()
    conn_devices_entry1.pack()

    Label(screen1, text="Select gender",font=("Times",13),bg="#FFFEF2").pack()

    gen_entry_m= Radiobutton(screen1, text="Male",variable=gender, value="Male",bg="#FFFEF2")
    gen_entry_m.deselect()
    gen_entry_m.pack()

    gen_entry_f= Radiobutton(screen1, text="Female",variable=gender, value="Female",bg="#FFFEF2")
    gen_entry_f.select()
    gen_entry_f.pack()

    Label(screen1, text="",bg="#FFFEF2").pack()
    Label(screen1, text="Enter Username",font=("Times",13),bg="#FFFEF2").pack()

    username_entry= Entry(screen1,textvariable=username, width=30)
    username_entry.pack()

    Label(screen1,text="Enter Password",font=("Times",13),bg="#FFFEF2").pack()

    password_entry= Entry(screen1,textvariable=password,width=30)
    password_entry.pack()


    Label(screen1,text="",bg="#FFFEF2").pack()
    Button(screen1, text="Register",bg="#d1ffea",height=2, width= 20, command= register_user).pack()

    Label(screen1,text="",bg="#FFFEF2").pack()
    Button(screen1, text="Direct to Login Page",bg="#d1ffea",height=2, width= 20, command= login).pack()

    Label(screen1,text="",bg="#FFFEF2").pack()
    Button(screen1,text="Close window",bg="#ffcccb",height=2, width= 20, command=delete_s1).pack()


#-----------------------------------------------------------DASHBOARD-----------------------------------------------------------------------

#these functions delete the screens
def delete_s4():
    screen4.destroy()
def delete_s5():
    screen5.destroy()
def delete_s8():
    screen8.destroy()
def delete_s9():
    screen9.destroy()

'''
#not fully complete
#purpose- saving the bluetooth info to new database

def save_info():
    conn= sqlite3.connect(resource_path("user_info.db"))
    c=conn.cursor()
    # c.execute("INSERT INTO Curr_session(bluetooth_name, bluetooth_address) VALUES('"+addr+"','"+name+"')")

    c.execute("UPDATE Curr_session SET bluetooth_name='"+addr+"',
    bluetooth_address='"+name+"' WHERE name='"+username1+"'")
    # messagebox.showinfo("Information","Your record is saved!")
    conn.commit()
    conn.close()
    conn= sqlite3.connect(resource_path("user_info.db"))
    c=conn.cursor()
    c.execute("SELECT * FROM Curr_session")
    r=c.fetchall()
    for i in r:
        print(i)
    # messagebox.showinfo("Information","Your record is saved!")
    conn.commit()
    conn.close()
'''

#Scanning nearby bluetooth devices

def button_clicked1():
    # print("success")
    # Label(screen9,text="hi").pack()

    #print("Scanning")
    Label(screen9,text="Scanning",width="400",height="3",font=("Times",13,"bold")).pack()

    nearby_devices = bluetooth.discover_devices(lookup_names=True)

    #print("Found {} devices.".format(len(nearby_devices)))
    Label(screen9,text="Found {} devices.".format(len(nearby_devices)),width="400",height="3",font=("Times",11,"bold")).pack()

    for addr, name in nearby_devices:
        #print("  {} - {}".format(addr, name))
        Label(screen9,text="  {} - {}".format(addr, name),width="400",height="3",font=("Times",12,"bold")).pack()


#Scan devices page

def scan_devices():
    global screen9
    screen9=Toplevel(screen)
    screen9.title("Scanning for nearby Bluetooth devices")
    screen9.geometry("400x500")
    Label(screen9,text="Nearby Bluetooth devices",font=("Times",16)).pack()
    Label(screen9,text="").pack()
    Button(screen9,text="Scan Devices", height=1, width= 30,
    bg="#0077b6",font=("Times",13,"bold"), command=button_clicked1).pack()
    # Button(screen9,text="Save", height=1, width= 30, bg="#CD5C5C",font=("Times",11,"bold"),command=save_info()).pack()
    #Label(screen9,text="hi").pack()
    #Label(screen9,text="").pack()
    Button(screen9,text="Exit", height=1, width= 30, bg="#CD5C5C",font=("Times",13,"bold"),command=delete_s9).pack(side=BOTTOM)



#Page for Dashboard

def session():
    # delete_s()
    delete_s2()
    global screen8
    screen8= Toplevel(screen)

    screen8.title("Dashboard")
    screen8.geometry("400x400")
    Label(screen8, text="Welcome to the Dashboard",font=("Times",16)).pack()

    Button(screen8,text="Sign out",bg="#ffcccb",command=delete_s8).place(x=340,y=10)

    conn= sqlite3.connect(resource_path("software.db"))
    c1=conn.cursor()
    c1.execute("SELECT name,gender FROM person WHERE username='"+username1+"'")
    r1=c1.fetchall()

    for i in r1:
        print(i[0])
        print(i[1])
        if(i[1]=="Female"):
            Label(screen8,text="Hello! Ms. {}".format(i[0]),font=("Times",14)).pack()
            # print("Ms. {}".format(i[0]))
        else:
            Label(screen8,text="Hello! Mr. {}".format(i[0]),font=("Times",14)).pack()
            # print("Mr. {}".format(i[0]))

    conn.commit()
    conn.close()

    Label(screen8,text="").pack()
    Button(screen8,text="Your Saved Devices",width=20,height=2,bg="#d1ffea").pack()

    Label(screen8,text="").pack()
    Button(screen8,text="Scan Nearby Devices",width=20,height=2,bg="#d1ffea",command=scan_devices).pack()
```

```
#-----------------------------------------------------------LOGIN SECTION-----------------------------------------------------------------

#if login is successful
def login_success():
    session()

#if passwoord is not recognised
def password_not_recognized():
    global screen4
    screen4 = Toplevel(screen)
    screen4.title("Failed")
    screen4.geometry("150x100")
    screen4.configure(bg="#ffcccb")
    Label(screen4,text="",bg="#ffcccb").pack()
    Label(screen4, text="Incorrect password",bg="#ffcccb",font=("Times",13)).pack()
    Button(screen4,text="Try again",bg="#ff6863",command=delete_s4).pack()

#to allow to go to registration page
def direct_register():
    register()
    delete_s5()

#if user is not found
def user_not_found():
    global screen5
    screen5 = Toplevel(screen)
    screen5.title("Failed")
    screen5.geometry("150x150")
    screen5.configure(bg="#ffdcd1")
    Label(screen5, text="User not found!",bg="#ffdcd1",font=("Times",13)).pack()
    Button(screen5,text="Try again",bg="#ff6863",command=delete_s5).pack()
    Label(screen5,text="",bg="#ffdcd1").pack()
    Button(screen5,text="Register new user",bg="#d1ffea",command=direct_register).pack()


#to check whether login credentials are correct or not
def login_verify():
    global username1
    global password1
    username1=username_verify.get()
    password1=password_verify.get()
    usernamee_entry1.delete(0,END)
    password_entry1.delete(0,END)

```

```python

    conn= sqlite3.connect(resource_path("software.db"))
    c=conn.cursor()
    c.execute("SELECT username,password FROM person")
    r=c.fetchall()

    user_list=[]
    pass_list=[]
    for i in r:
        user_list.append(i[0])
        pass_list.append(i[1])
    print(user_list)
    print(pass_list)
    for i in range(len(user_list)):
        if(username1 in user_list):
            if(password1 in pass_list):
                idx=user_list.index(username1)
                if(pass_list[idx]==password1):
                    # print("YES")
                    login_success()
                    break
                else:
                    # print("Incorrect Password")
                    messagebox.askretrycancel("Try Again","Password Incorrect")
                    break
            else:
                # print("incorrect password")
                messagebox.askretrycancel("Try Again","Password Incorrect")
                break
        else:
            # print("User Not Found")
            user_not_found()
            break

    conn.commit()
    conn.close()


def delete_s2():
    screen2.destroy()

#The login page
def login():
    print("Login session started")
    global screen2
    screen2=Toplevel(screen)
    # screen2=Tk()
    screen2.title("Login page")
    screen2.geometry("400x350")
    screen2.configure(bg="#E5F6DF")
    Label(screen2,text="Please enter details below to login",width="400",height="3",bg="#b6e9f2", font=("Times",16)).pack()
    Label(screen2,text="",bg="#E5F6DF").pack()

    global username_verify
    global password_verify
    username_verify= StringVar()
    password_verify=StringVar()
    Label(screen2,text="Enter your registered Username",font=("Times",13),bg="#E5F6DF").pack()
    global usernamee_entry1
    global password_entry1
    usernamee_entry1= Entry(screen2, textvariable=username_verify,width=30)
    usernamee_entry1.pack()
    Label(screen2,text="Enter your Password",font=("Times",13),bg="#E5F6DF").pack()
    password_entry1=Entry(screen2,textvariable=password_verify, show="*",width=30)
    password_entry1.pack()
    Label(screen2,text="",bg="#E5F6DF").pack()
    Button(screen2,text="Login",width=20,height=2, command=login_verify,bg="#d1ffea").pack()
    Label(screen2,text="",bg="#E5F6DF").pack()
    Button(screen2,text="Close window",bg="#ffcccb",height=2, width= 20, command=delete_s2).pack()


#------------------------------------------------------------MAIN SCREEN-------------------------------------------------------------------------

def main_screen():

    global screen
    screen=Tk()
    screen.geometry("400x400")
    screen.minsize(400,400)
    screen.maxsize(400,400)

    #QUERIES to create databases(implemented for only once)
    # conn= sqlite3.connect("software.db")
    # c=conn.cursor()
    # c.execute("CREATE TABLE person(name TEXT, age INT, conn_type TEXT, conn_type_other TEXT, gender TEXT, username TEXT, password TEXT)")
    # conn.commit()
    # conn.close()
    # conn= sqlite3.connect("user_info.db")
    # c=conn.cursor()
    # c.execute("CREATE TABLE Curr_session(name TEXT, bluetooth_name TEXT,
    bluetooth_address TEXT)")
    # conn.commit()
    # conn.close()

    screen.title("Sign up/ Login page")
    screen.configure(bg="#caf0f8")
    Label(text="Sign up/ Log in",width="400",height="3", fg="#caf0f8",bg="#04035e", font=("Times",16,"bold")).pack()
    photo = PhotoImage(file=resource_path("image.png"), width="400", height="150")
    new=Label(image=photo)
    new.pack()
    Label(text="",bg="#caf0f8").pack()
    Button(text="Login", height=1, width= 30, bg="#0077b6",font=("Times",13,"bold"), command=login).pack()
    Label(text="",bg="#caf0f8").pack()
    Button(text="Register", height=1, width= 30,bg="#0077b6",font=("Times",13,"bold"), command=register).pack()
    Label(text="",bg="#caf0f8").pack()
    Button(text="Quit", height=1, width= 30,bg="#0077b6",font=("Times",13,"bold"), command=lambda: screen.quit()).pack()
    screen.mainloop()

main_screen()

```