

```

1  #-----IMPORT
LIBRARIES-----
2
3  from tkinter import *
4  import tkinter.font as font
5  from PIL import Image, ImageTk
6  import sqlite3
7  from tkinter import messagebox
8  import serial
9  import bluetooth
10 import sys
11 import os
12
13 #for converting to windows software
14 def resource_path(relative_path):
15     """ Get absolute path to resource, works for dev and for PyInstaller """
16     try:
17         # PyInstaller creates a temp folder and stores path in _MEIPASS
18         base_path = sys._MEIPASS2
19     except Exception:
20         base_path = os.path.abspath(".")
21
22     return os.path.join(base_path, relative_path)
23
24
25 #-----REGISTRATION
SECTION-----
26
27 #saves the registered values into the database
28 def register_user():
29     name_info=name.get()
30     age_info=age.get()
31     conn_type_of_devices_info=conn_type_of_devices.get()
32     conn_type_of_devices_info_1=conn_type_of_devices_1.get()
33     gender_info=gender.get()
34     username_info=username.get()
35     password_info= password.get()
36
37     #software.db contains registered info
38     conn= sqlite3.connect(resource_path("software.db"))
39     c=conn.cursor()
40     c.execute("INSERT INTO person
VALUES('"+name_info+"','"+age_info+"','"+conn_type_of_devices_info+"','"+conn_type_o
f_devices_info_1+"','"+gender_info+"','"+username_info+"','"+password_info+"')")
41     messagebox.showinfo("Information","Your record is saved!")

```

```
42     conn.commit()
43     conn.close()
44     conn= sqlite3.connect(resource_path("user_info.db"))
45     c=conn.cursor()
46     c.execute("INSERT INTO Curr_session(name) VALUES('"+name_info+"')")
47     conn.commit()
48     conn.close()
49
50     name_entry.delete(0,END)
51     age_entry.delete(0,END)
52     conn_devices_entry.deselect()
53     conn_devices_entry1.deselect()
54     gen_entry_m.deselect()
55     gen_entry_f.select()
56     username_entry.delete(0,END)
57     password_entry.delete(0,END)
58
59     Label(screen1,text="Registration is Successful", fg="green",
60           font=("Calibri",11)).pack()
61
62     #these functions delete the screen
63     def delete_s1():
64         screen1.destroy()
65
66     #for the registration page
67     def register():
68         global screen1
69         screen1= Toplevel(screen)
70         screen1.title("Register")
71         screen1.geometry("400x650")
72         screen1.configure(bg="#FFFEF2")
73
74         global name
75         global age
76         global conn_type_of_devices
77         global conn_type_of_devices_1
78         global gender
79         global username
80         global password
81
82         global name_entry
83         global age_entry
84         global conn_devices_entry
85         global conn_devices_entry1
```

2

```

86     global gen_entry_m
87     global gen_entry_f
88     global username_entry
89     global password_entry
90
91     name=StringVar()
92     age=StringVar()
93     conn_type_of_devices=StringVar()
94     conn_type_of_devices_1=StringVar()
95     gender=StringVar()
96     username= StringVar()
97     password= StringVar()
98
99     Label(screen1, text="Please Enter details below
    ",width="400",height="3",bg="#caf0f8", font=("Times",16)).pack()
100
101     Label(screen1, text="",bg="#FFFEF2").pack()
102
103     Label(screen1, text="Enter Name",font=("Times",13),bg="#FFFEF2").pack()
104
105     name_entry= Entry(screen1,textvariable=name, width=30)
106     name_entry.pack()
107
108     Label(screen1, text="Enter Age",font=("Times",13),bg="#FFFEF2").pack()
109
110     age_entry= Entry(screen1,textvariable=age, width=30)
111     age_entry.pack()
112
113     Label(screen1, text="Select types of
    connection",font=("Times",13),bg="#FFFEF2").pack()
114
115     conn_devices_entry= Checkbutton(screen1,
    text="Wifi",variable=conn_type_of_devices,bg="#FFFEF2")
116     conn_devices_entry.deselect()
117     conn_devices_entry.pack()
118
119     conn_devices_entry1= Checkbutton(screen1,
    text="Bluetooth",variable=conn_type_of_devices_1,bg="#FFFEF2")
120     conn_devices_entry1.deselect()
121     conn_devices_entry1.pack()
122
123     Label(screen1, text="Select gender",font=("Times",13),bg="#FFFEF2").pack()
124
125     gen_entry_m= Radiobutton(screen1, text="Male",variable=gender,
    value="Male",bg="#FFFEF2")

```

```

126     gen_entry_m.deselect()
127     gen_entry_m.pack()
128
129     gen_entry_f= Radiobutton(screen1, text="Female",variable=gender,
130     value="Female",bg="#FFFEF2")
131     gen_entry_f.select()
132     gen_entry_f.pack()
133
134     Label(screen1, text="",bg="#FFFEF2").pack()
135     Label(screen1, text="Enter Username",font=("Times",13),bg="#FFFEF2").pack()
136
137     username_entry= Entry(screen1,textvariable=username, width=30)
138     username_entry.pack()
139
140     Label(screen1,text="Enter Password",font=("Times",13),bg="#FFFEF2").pack()
141
142     password_entry= Entry(screen1,textvariable=password,width=30)
143     password_entry.pack()
144
145     Label(screen1,text="",bg="#FFFEF2").pack()
146     Button(screen1, text="Register",bg="#d1ffea",height=2, width= 20, command=
147     register_user).pack()
148
149     Label(screen1,text="",bg="#FFFEF2").pack()
150     Button(screen1, text="Direct to Login Page",bg="#d1ffea",height=2, width= 20,
151     command= login).pack()
152
153     Label(screen1,text="",bg="#FFFEF2").pack()
154     Button(screen1,text="Close window",bg="#ffcccb",height=2, width= 20,
155     command=delete_s1).pack()
156
157     #-----DASHBOARD-----
158
159     #these functions delete the screens
160     def delete_s4():
161         screen4.destroy()
162     def delete_s5():
163         screen5.destroy()
164     def delete_s8():
165         screen8.destroy()
166     def delete_s9():
167         screen9.destroy()

```

```

166
167     '''
168     #not fully complete
169     #purpose- saving the bluetooth info to new database
170
171     def save_info():
172         conn= sqlite3.connect(resource_path("user_info.db"))
173         c=conn.cursor()
174         # c.execute("INSERT INTO Curr_session(bluetooth_name, bluetooth_address)
VALUES('"+addr+"', '"+name+"')")
175
176         c.execute("UPDATE Curr_session SET bluetooth_name='"+addr+"',
bluetooth_address='"+name+"' WHERE name='"+username1+"'")
177         # messagebox.showinfo("Information","Your record is saved!")
178         conn.commit()
179         conn.close()
180         conn= sqlite3.connect(resource_path("user_info.db"))
181         c=conn.cursor()
182         c.execute("SELECT * FROM Curr_session")
183         r=c.fetchall()
184         for i in r:
185             print(i)
186         # messagebox.showinfo("Information","Your record is saved!")
187         conn.commit()
188         conn.close()
189     '''
190
191     #Scanning nearby bluetooth devices
192
193     def button_clicked1():
194         # print("success")
195         # Label(screen9,text="hi").pack()
196
197         #print("Scanning")
198         Label(screen9,text="Scanning",width="400",height="3",
font=("Times",13,"bold")).pack()
199
200         nearby_devices = bluetooth.discover_devices(lookup_names=True)
201
202         #print("Found {} devices.".format(len(nearby_devices)))
203         Label(screen9,text="Found {}
devices.".format(len(nearby_devices)),width="400",height="3",
font=("Times",11,"bold")).pack()
204
205         for addr, name in nearby_devices:

```

```

206         #print(" {} - {}".format(addr, name))
207         Label(screen9,text=" {} - {}".format(addr, name),width="400",height="3",
                font=("Times",12,"bold")).pack()
208
209
210 #Scan devices page
211
212 def scan_devices():
213     global screen9
214     screen9=Toplevel(screen)
215     screen9.title("Scanning for nearby Bluetooth devices")
216     screen9.geometry("400x500")
217     Label(screen9,text="Nearby Bluetooth devices",font=("Times",16)).pack()
218     Label(screen9,text="").pack()
219     Button(screen9,text="Scan Devices", height=1, width= 30,
                bg="#0077b6",font=("Times",13,"bold"), command=button_clicked1).pack()
220     # Button(screen9,text="Save", height=1, width= 30,
                bg="#CD5C5C",font=("Times",11,"bold"),command=save_info()).pack()
221     #Label(screen9,text="hi").pack()
222     #Label(screen9,text="").pack()
223     Button(screen9,text="Exit", height=1, width= 30,
                bg="#CD5C5C",font=("Times",13,"bold"),command=delete_s9).pack(side=BOTTOM)
224
225
226
227 #Page for Dashboard
228
229 def session():
230     # delete_s()
231     delete_s2()
232     global screen8
233     screen8= Toplevel(screen)
234
235     screen8.title("Dashboard")
236     screen8.geometry("400x400")
237     Label(screen8, text="Welcome to the Dashboard",font=("Times",16)).pack()
238
239     Button(screen8,text="Sign out",bg="#ffcccb",command=delete_s8).place(x=340,y=10)
240
241     conn= sqlite3.connect(resource_path("software.db"))
242     c1=conn.cursor()
243     c1.execute("SELECT name,gender FROM person WHERE username='"+username1+"'")
244     r1=c1.fetchall()
245
246     for i in r1:

```

```

247         print(i[0])
248         print(i[1])
249         if(i[1]=="Female"):
250             Label(screen8,text="Hello! Ms. {}".format(i[0]),font=("Times",14)).pack()
251             # print("Ms. {}".format(i[0]))
252         else:
253             Label(screen8,text="Hello! Mr. {}".format(i[0]),font=("Times",14)).pack()
254             # print("Mr. {}".format(i[0]))
255
256     conn.commit()
257     conn.close()
258
259     Label(screen8,text="").pack()
260     Button(screen8,text="Your Saved Devices",width=20,height=2,bg="#d1ffea").pack()
261
262     Label(screen8,text="").pack()
263     Button(screen8,text="Scan Nearby
Devices",width=20,height=2,bg="#d1ffea",command=scan_devices).pack()
264
265
266     #-----LOGIN
SECTION-----
267
268     #if login is successful
269     def login_success():
270         session()
271
272     #if password is not recognised
273     def password_not_recognized():
274         global screen4
275         screen4 = Toplevel(screen)
276         screen4.title("Failed")
277         screen4.geometry("150x100")
278         screen4.configure(bg="#ffcccb")
279         Label(screen4,text="",bg="#ffcccb").pack()
280         Label(screen4, text="Incorrect password",bg="#ffcccb",font=("Times",13)).pack()
281         Button(screen4,text="Try again",bg="#ff6863",command=delete_s4).pack()
282
283     #to allow to go to registration page
284     def direct_register():
285         register()
286         delete_s5()
287
288     #if user is not found
289     def user_not_found():

```

```

290     global screen5
291     screen5 = Toplevel(screen)
292     screen5.title("Failed")
293     screen5.geometry("150x150")
294     screen5.configure(bg="#ffdcdd")
295     Label(screen5, text="User not found!", bg="#ffdcdd", font=("Times", 13)).pack()
296     Button(screen5, text="Try again", bg="#ff6863", command=delete_s5).pack()
297     Label(screen5, text="", bg="#ffdcdd").pack()
298     Button(screen5, text="Register new
user", bg="#d1ffea", command=direct_register).pack()

299
300
301 #to check whether login credentials are correct or not
302 def login_verify():
303     global username1
304     global password1
305     username1=username_verify.get()
306     password1=password_verify.get()
307     usernamee_entry1.delete(0,END)
308     password_entry1.delete(0,END)
309
310
311     conn= sqlite3.connect(resource_path("software.db"))
312     c=conn.cursor()
313     c.execute("SELECT username,password FROM person")
314     r=c.fetchall()
315
316     user_list=[]
317     pass_list=[]
318     for i in r:
319         user_list.append(i[0])
320         pass_list.append(i[1])
321     print(user_list)
322     print(pass_list)
323     for i in range(len(user_list)):
324         if(username1 in user_list):
325             if(password1 in pass_list):
326                 idx=user_list.index(username1)
327                 if(pass_list[idx]==password1):
328                     # print("YES")
329                     login_success()
330                     break
331             else:
332                 # print("Incorrect Password")
333                 messagebox.askretrycancel("Try Again", "Password Incorrect")

```



```

334             break
335         else:
336             # print("incorrect password")
337             messagebox.askretrycancel("Try Again", "Password Incorrect")
338             break
339     else:
340         # print("User Not Found")
341         user_not_found()
342         break
343
344     conn.commit()
345     conn.close()
346
347
348 def delete_s2():
349     screen2.destroy()
350
351 #The login page
352 def login():
353     print("Login session started")
354     global screen2
355     screen2=Toplevel(screen)
356     # screen2=Tk()
357     screen2.title("Login page")
358     screen2.geometry("400x350")
359     screen2.configure(bg="#E5F6DF")
360     Label(screen2,text="Please enter details below to
login",width="400",height="3",bg="#b6e9f2", font=("Times",16)).pack()
361     Label(screen2,text="",bg="#E5F6DF").pack()
362
363     global username_verify
364     global password_verify
365     username_verify= StringVar()
366     password_verify=StringVar()
367     Label(screen2,text="Enter your registered
Username",font=("Times",13),bg="#E5F6DF").pack()
368     global usernamee_entry1
369     global password_entry1
370     usernamee_entry1= Entry(screen2, textvariable=username_verify,width=30)
371     usernamee_entry1.pack()
372     Label(screen2,text="Enter your Password",font=("Times",13),bg="#E5F6DF").pack()
373     password_entry1=Entry(screen2,textvariable=password_verify, show="*",width=30)
374     password_entry1.pack()
375     Label(screen2,text="",bg="#E5F6DF").pack()
376     Button(screen2,text="Login",width=20,height=2,

```

```

        command=login_verify,bg="#d1ffea").pack()
377     Label(screen2,text="",bg="#E5F6DF").pack()
378     Button(screen2,text="Close window",bg="#ffcccb",height=2, width= 20,
        command=delete_s2).pack()

379
380
381     #-----MAIN
SCREEN-----

382
383     def main_screen():
384
385         global screen
386         screen=Tk()
387         screen.geometry("400x400")
388         screen.minsize(400,400)
389         screen.maxsize(400,400)
390
391         #QUERIES to create databases(implemented for only once)
392         # conn= sqlite3.connect("software.db")
393         # c=conn.cursor()
394         # c.execute("CREATE TABLE person(name TEXT, age INT, conn_type TEXT,
        conn_type_other TEXT, gender TEXT, username TEXT, password TEXT)")
395         # conn.commit()
396         # conn.close()
397         # conn= sqlite3.connect("user_info.db")
398         # c=conn.cursor()
399         # c.execute("CREATE TABLE Curr_session(name TEXT, bluetooth_name TEXT,
        bluetooth_address TEXT)")
400         # conn.commit()
401         # conn.close()
402
403         screen.title("Sign up/ Login page")
404         screen.configure(bg="#caf0f8")
405         Label(text="Sign up/ Log in",width="400",height="3", fg="#caf0f8",bg="#04035e",
        font=("Times",16,"bold")).pack()
406         photo = PhotoImage(file=resource_path("image.png"), width="400", height="150")
407         new=Label(image=photo)
408         new.pack()
409         Label(text="",bg="#caf0f8").pack()
410         Button(text="Login", height=1, width= 30, bg="#0077b6",font=("Times",13,"bold"),
        command=login).pack()
411         Label(text="",bg="#caf0f8").pack()
412         Button(text="Register", height=1, width=
        30,bg="#0077b6",font=("Times",13,"bold"), command=register).pack()
413         Label(text="",bg="#caf0f8").pack()

```

```
414         Button(text="Quit", height=1, width= 30,bg="#0077b6",font=("Times",13,"bold"),
415               command=lambda: screen.quit()).pack()
416         screen.mainloop()
417     main_screen()
418
```