

Animal_Taxonomy_Ctk_Main.py

```

1  from customtkinter import *
2  from subprocess import call
3  from PIL import Image
4  import os
5  import sqlite3
6  from tkinter import messagebox
7  from Global_Config import *
8
9  root = CTk()
10
11  global glb_color_1, glb_color_2, glb_color_3
12
13  glb_color_1 = "darkorchid2"# #FFC125 #FFCC70
14  glb_color_2 = "dodgerblue3"
15  glb_color_3 = "#308014"# darkorchid2, #308014 #c850c0
16
17  centreScreen(root, root,600,600)
18  root.title("Animal Taxonomy")
19  root.maxsize(width = 600, height = 600)
20
21  root.iconbitmap(r"icon/favicon6.ico")
22  set_appearance_mode("Dark")
23
24  con = sqlite3.connect("Animal_Taxonomy_Db.db")
25  cur = con.cursor()
26
27  global glb_current_working_directory
28  glb_current_working_directory = os.path.dirname(os.path.realpath(__file__))
29
30  def validation():
31      username = username_entry.get()
32      password = password_entry.get()
33      if username == "" or password == "":
34          error_label = CTKLabel(login_frame, text = "Username or password is wrong", font
= ("Bradley Hand ITC" , 15, "italic", "bold"), text_color = "red")
35          error_label.place(x = 75, y = 170)
36          def refresh():
37              error_label.destroy()
38              error_label.after(3000, refresh)
39
40          # username_entry.configure(border_color = "red")
41          # password_entry.configure(border_color = "red")
42      else :
43          tmp_qry = "SELECT Username, Password FROM User_details WHERE Username = '"+username+"
' AND Password = '"+password+"' AND Active = 1"
44          cur.execute(tmp_qry)
45          row = cur.fetchone()
46          if row :
47              def admin_console():
48                  login.destroy()
49                  call(["python", glb_current_working_directory + "
/Animal_Taxonomy_CTk_Admin_Console.py"])
50                  admin_console()
51          else:

```

```

52         error_label = CTKLabel(login_frame, text = "Username or password is wrong", font
= ("Bradley Hand ITC" , 15, "italic", "bold"), text_color = "red")
53         error_label.place(x = 75, y = 170)
54         def refresh():
55             error_label.destroy()
56             error_label.after(3000, refresh)
57
58 welcome_text = "WELCOME"
59
60 def redirect_to_user(_isadmin = False):
61     if _isadmin:
62         login_text = "LOGIN"
63         global login
64         login = CTK()
65         login.iconbitmap(r"icon/favicon6.ico")
66         centreScreen(login, root,400,200)
67         login.title("Admin Login")
68         login.maxsize(width = 400, height = 200)
69
70         global login_frame
71         login_frame = CTKFrame(login, border_color = glb_color_1, border_width = 2, width =
400, height = 200)
72         login_frame.pack()
73
74         login_label = CTKLabel(login_frame, text = login_text, font = ("Bradley Hand ITC" ,
40, "italic", "bold"), text_color = glb_color_3)
75         login_label.place(x = 130, y = 5)
76
77         username_label = CTKLabel(login_frame, text = "Username :-", font = ("Bradley Hand
ITC" , 20, "italic", "bold"), text_color = glb_color_2)
78         username_label.place(x = 15, y = 70)
79
80         global username_entry
81         username_entry = CTKEntry(login_frame, text_color = glb_color_3, width = 200)
82         username_entry.place(x = 170, y = 70)
83
84         password_label = CTKLabel(login_frame, text = "Password:-", font = ("Bradley Hand
ITC" , 20, "italic", "bold"), text_color = glb_color_2)
85         password_label.place(x = 15, y = 110)
86
87         global password_entry
88         password_entry = CTKEntry(login_frame, show = "*", text_color = glb_color_3, width =
200)
89         password_entry.place(x = 170, y = 110)
90
91         submit_btn = CTKButton(login_frame, height = 15, text = "Submit", fg_color =
glb_color_2,hover_color = glb_color_3,corner_radius = 35,
92             command = lambda: validation())
93         submit_btn.place(x = 45, y = 150)
94
95         def back_to_main_console():
96             login.destroy()
97             call(["python", glb_current_working_directory + "/Animal_Taxonomy_Ctk_Main.py"])
98
99         cancel_btn = CTKButton(login_frame, height = 15, text = "Back", fg_color =
glb_color_2,hover_color = glb_color_3,corner_radius = 35,
100             command = lambda: (back_to_main_console()))
101         cancel_btn.place(x = 210, y = 150)

```

```
102
103     root.destroy()
104     login.mainloop()
105
106     else:
107         def guest_console():
108             root.destroy()
109             call(["python", glb_current_working_directory + "
/Animal_Taxonomy_CTk_Guest_Console.py"])
110             guest_console()
111
112 content_frame = CTKFrame(root, border_color = glb_color_1, border_width = 2, width = 600,
height = 600)
113
114 welcome_message = CTKLabel(content_frame, text = welcome_text, font = ("Brush Script MT" ,
50, "italic" ))
115 welcome_message.place(x = (600/2-len(welcome_text)//2)-100, y = 200)
116
117 welcome_message = CTKLabel(content_frame, text = "To the world of animals", font = ("Brush
Script MT" , 18, "italic" ))
118 welcome_message.place(x = (600/2-len(welcome_text)//2)-50, y = 280)
119
120 guest_mode_btn = CTKButton(content_frame, text = "View As A Guest...", fg_color =
glb_color_2, hover_color = glb_color_3, corner_radius = 35,
121                             width = 240, command = lambda :(redirect_to_user()))
122 guest_mode_btn.place(x = 45, y = 330)
123
124 admin_mode_btn = CTKButton(content_frame, text = "View As An Admin...", fg_color =
glb_color_2, hover_color = glb_color_3, corner_radius = 35,
125                             width = 240, command = lambda :(redirect_to_user(True)))
126 admin_mode_btn.place(x = 320, y = 330)
127
128 content_frame.place(x = 0, y = 0)
129
130 root.mainloop()
131
```