

fkos.py

```
1
2 from tkinter import*
3 from tkinter import messagebox
4 import os
5 from math import*
6 import urllib.parse
7 import webbrowser
8 import time
9
10
11 # #6be4ce light colour      #21d1b3  dark colour      wallpaper colours
12 # #007166  default theme
13
14
15 def confirmext():
16     if messagebox.askyesno("Exit", "Do you really want to quit?"):
17         win.destroy()
18
19
20 def global_click_handler(event):    #handles all clicks #removedclickin out and checkin
    out for better usage and functionality in this code
21     widget = event.widget
22
23     if widget != searchbar:
24         if searchbar.get() == '':
25             searchbar.insert(0, 'Search Anything')
26             searchbar.config(fg='gray')
27             win.focus_set()
28     else:
29         if searchbar.get() == 'Search Anything':
30             searchbar.delete(0, END)
31             searchbar.config(fg='black')
32     if widget != signin:
33         if signin.get() == '':
34             signin.insert(0, 'Password')
35             signin.config(fg='gray', show='')
36     else:
37         if signin.get() == 'Password':
38             signin.delete(0, END)
39             signin.config(fg='black', show='*')
40     if widget not in (signin, searchbar):
41         win.focus_set()
42
43
44
45 def delsear():
46     searchbar.delete(0,END)
47     searchbar.insert(0,'Search Anything')
48     searchbar.config(fg="gray")
49     win.focus()
50 def search():
51     query = searchbar.get()
```

```

52     if query and query != 'Search Anything':
53         try:
54             url = "https://www.google.com/search?q=" + urllib.parse.quote(query)
#convert the search in a+b+c format more url friendly
55             webbrowser.open_new_tab(url)         #open in new tab not new chrome
56         except Exception as e:
57             messagebox.showerror("Error", f"Could not open browser:\n{e}")
58     else:
59         messagebox.showerror("Error", 'Search something')
60
61
62 count=0
63 def checkpass():
64     global count
65     count+=1
66     passw=signin.get()
67     if passw=='1234':
68         signif.destroy()
69         entryscreen.update()
70         time.sleep(2)
71         signif.destroy()
72         entryscreen.update()
73         entryscreen.destroy()
74     else:
75         if count>2:
76             turnoff()
77         else:
78             signin.delete(0,END)
79             messagebox.showerror("Wrong Password",f'Enter correct password {3-count} tries
left')
80 def turnoff():
81     messagebox.showerror("Turning off", 'maximum amount of tries used')
82     quit()
83
84
85
86 def calculate():
87     ste=StringVar()
88     ste.set('')
89     global calpic
90
91     def click(event):
92         t = event.widget.cget('text')
93         current = ste.get()
94
95         try:
96             if t == '=':
97                 expr = current.replace('x', '*').replace('÷', '/')
98                 result = eval(expr)
99                 ste.set(str(result))
100
101             elif t in ['C', 'CE']:
102                 ste.set('')
103

```

```
104         elif t == '⊗':
105             ste.set(current[:-1])
106             screen.update()
107
108         elif t == '1/x':
109             if current:
110                 result = 1 / float(current)
111                 ste.set(str(result))
112                 screen.update()
113
114         elif t == 'x²':
115             if current:
116                 result = float(current) ** 2
117                 ste.set(str(result))
118                 screen.update()
119
120         elif t == '√x':
121             if current:
122                 num = float(current)
123                 if num < 0:
124                     ste.set("Error")
125                     screen.update()
126                 else:
127                     ste.set(str(num ** 0.5))
128                     screen.update()
129
130         elif t == '+/-':
131             if current:
132                 num = float(current)
133                 ste.set(str(-num))
134                 screen.update()
135
136         elif t == '%':
137             if current:
138                 num = float(current)
139                 ste.set(str(num / 100))
140                 screen.update()
141
142         else:
143             ste.set(current + t)
144             screen.update()
145
146     except Exception as e:
147         ste.set("Error")
148         screen.update()
149
150 def click_simulate_equal():
151     expr = ste.get().replace('x', '*').replace('÷', '/')
152     try:
153         result = eval(expr)
154         ste.set(str(result))
155         screen.update()
156     except:
157         ste.set("Error")
```

```

158         screen.update()
159
160
161
162     calc = Toplevel()
163     calc.iconphoto(True,calpic)
164     calc.title("Calculator")
165     calc.config(bg="#00C4B0")
166     calc.geometry("+800+350")
167
168     screen = Entry(calc, font='lucida 28 bold', textvariable=ste,
169 justify=RIGHT,bg="#91FFF4",fg="#001D1A")
170     screen.config(width=12)
171     screen.grid(row=0, column=0, columnspan=4)
172
173     buttons = [
174         ['%', 'CE', 'C', '⊗'],
175         ['1/x', 'x²', '√x', '÷'],
176         ['7', '8', '9', 'x'],
177         ['4', '5', '6', '-'],
178         ['1', '2', '3', '+'],
179         ['+/-', '0', '.', '=']
180     ]
181
182     for i in range(len(buttons)):
183         for j in range(len(buttons[i])):
184             b = Button(calc, text=buttons[i][j], font='lucida 20 bold', relief='flat',
185 bd=0, highlightthickness=0,
186 bg="#00C4B0",activebackground="#00C4B0",fg="#00312C",activeforeground='#00574E')
187             b.grid(row=i+1, column=j)
188             b.bind("<Button-1>", click)
189             screen.focus_set()
190             calc.bind("<Return>", lambda event: click_simulate_equal())
191
192
193
194
195 def fundang():    #afun easter egg just for whow may delete in future
196     dangwin=Toplevel()
197     dangwin.geometry("800x500+300+150")
198     dangwin.config(bg="#007166")
199     danglab=Label(dangwin,image=dangerpic,bg='#007166')
200     danglab2=Label(dangwin,text="WHY DID YOU OPENED IT ",font=(30),bg='#007166')
201     danglab3=Label(dangwin,text="WHY WHY WHY ... ",font=(30),bg='#007166')
202     danglab4=Label(dangwin,text="WHY YOU KILLED ME ",font=(30),bg='#007166')
203     danglab.pack(fill=BOTH)
204     danglab2.pack(fill=BOTH)
205     danglab3.pack(fill=BOTH)
206     danglab4.pack(fill=BOTH)
207

```

```

208     dangwin.update() # Force window to render before freezing bcs as soon as fucntion is
loaded it frezzes bcs of sleep
209     time.sleep(5)
210     quit()
211
212
213 def open_folder(path): #open actual path of folder
214     explorer = Toplevel()
215     explorer.title(f"Explorer - {os.path.basename(path)}") # Set window title as folder
name f'{x}' is use to incoperate variable in strig
216     explorer.geometry("800x500+300+150") # Size and position
217     explorer.config(bg="#007166") # FKOS theme
218     explorer.iconphoto(True, mypcpic) # Same icon as "My PC"
219     current_path = StringVar(value=path) # Track current folder path does like intvar
stringvar is use to store dynamic text
220
221     listbox=Listbox(explorer,bg="#007166",font=('Segoe
UI',18,'bold'),selectbackground='#21d1b3',relief="flat", highlightthickness=0, bd=0)
222     listbox.pack(fill=BOTH, expand=True, padx=10, pady=10)
223     #fill=both will allow to listbox to expand in both side in window
224     #expand =true allow listbox to take as much space as required
225     #pad x/y adds padding as before
226
227
228     def list_folder(p):
229         try:
230             files=os.listdir(p) #get the content of folder
231         except:
232             messagebox.showerror("ERROR",f"file {p} not found")
#messagebox.showerror("title","message")
233         return
234         listbox.delete(0,END) #clear old entries
235         for f in files:
236             listbox.insert(END,f)
237         current_path.set(p) #update tracker to show which folder you re in
238         win.config(cursor='dotbox')
239
240
241     def double_click(event):
242         selected=listbox.get(listbox.curselection()) #what user double click
243         new_path=os.path.join(current_path.get(),selected) #create full path
244
245         if os.path.isdir(new_path): # checks if its a folder and if it is go inside
246             list_folder(new_path)
247         else:
248             try:
249                 os.startfile(new_path) #if files open by default
250             except:
251                 messagebox.showerror("ERROR",f"File {new_path} can't be accessed")
252         listbox.bind("<Double-Button-1>",double_click)
253         list_folder(path)
254
255
256 def mypc():

```

```

257
258     # i have used file directory preferable to me please change the directory to you
desired folder
259     # floder a published alongside have various drafts of my code
260     # you can easily see how to make it yourself in case you want to
261
262     pc=Toplevel()
263     fpic_local=PhotoImage(file='folder 100.png').zoom(2,2)
264     pc.config(background='#007166')
265     pc.geometry('1200x720+200+100')
266     pc.title('My pc')
267     pc.iconphoto(True,mycpic)
268     folA=Button(pc,
269                 image=fpic_local,
270                 bg="#007166",activebackground='#007166',
271                 relief='flat',bd=0,highlightthickness=0,
272                 command=lambda:open_folder(r'F:\fkos\folder A'),
273                 text='Folder A',fg='white',font=('Segoe UI',18,'bold'),compound='top',)
#open actual folder    r'F:/folderA' is use bcs \f is escape command
274     folA.image = fpic_local    #preserve the image
275     folA.place(x=45,y=30)
276     folB=Button(pc,
277                 image=fpic_local,
278                 bg="#007166",activebackground='#007166',
279                 relief='flat',bd=0,highlightthickness=0,
280                 command=lambda:open_folder(r'F:\fkos\folder B'),
281                 text='Folder B',fg='white',font=('Segoe UI',18,'bold'),compound='top')
#open actual folder    r'F:/folderA' is use bcs \f is escape command
282     folB.image = fpic_local    #preserve the image
283     folB.place(x=345,y=30)
284
285
286
287 win=Tk()
288
289 walp=PhotoImage(file="wallpaper.png")
290 mycpic=PhotoImage(file='my pc.png')
291 calpic=PhotoImage(file="calculator.png")
292 searchpic=PhotoImage(file='search40.png')
293 dangerpic=PhotoImage(file="danger100.png")
294 userpic=PhotoImage(file='user logo.png')
295
296 win.attributes('-fullscreen',True)
297 wallpaper=Label(image=walp)
298 wallpaper.place(x=0,y=0)
299 icon=PhotoImage(file='logo.png')
300 win.iconphoto(True,icon)
301 win.title("FKOS")
302 tpc=Button(image=mycpic,bg='#6be4ce',activebackground='#6be4ce',
303            relief='flat',bd=0,highlightthickness=0,            #for making button flat
304            text='My pc',font=('Segoe UI',9,'bold'),compound='top',
305            command=mypc)
306 tpc.place(x=15,y=25)
307

```

```

308
309 tcalc=Button(image=calpic,bg='#6be4ce',activebackground='#6be4ce',
310             relief='flat',bd=0,highlightthickness=0,           #for making button flat
311             text='Calculator',font=('Segoe UI',9,'bold'),compound='top',command=calculate)
312 tcalc.place(x=15,y=150)
313
314
315 dang=Button(win,image=dangerpic,text="DON'T OPEN",font=('Segoe UI',9,'bold'),
316             relief='flat',bd=0,highlightthickness=0,bg='#6be4ce',
317             activebackground='#6be4ce',compound='top',command=fundang)
318 dang.place(x=15,y=275)
319
320
321 taskbar1=Label(win,bg="#37FFEB",height=37,width=130)
322 taskbar1.place(x=0,y=1043)
323
324 taskbar2=Label(win,bg="#37FFEB",height=37,width=145)
325 taskbar2.place(x=1000,y=1043)
326
327 power_logo=icon.subsample(2,2)    #logo is 100x100
328 powerbut=Button(win,image=power_logo,relief='flat',bd=0,highlightthickness=0,
329                 bg='#37FFEB',activebackground='#37FFEB',command=confirmext)
330 powerbut.place(x=690, y=1043)
331
332 searchbar=Entry(win,font=('Segoe UI',20,),fg="gray",
333                 relief='flat',bd=0,highlightthickness=0)
334 searchbar.insert(0,'Search Anything')
335 searchbar.config(width=23)
336 searchbar.place(x=740,y=1043)
337
338 delsearch=Button(win,text='X',font=("Verdana", 20, "bold"),fg="gray",
339                 bg='FFFFFF',activebackground='FFFFFF',
340                 relief='flat',bd=0,highlightthickness=0,command=delsear)
341 delsearch.place(x=1086,y=1043)
342
343 searchbut=Button(win,image=searchpic,bg='#37FFEB',activebackground='#37FFEB',
344                 relief='flat',bd=0,highlightthickness=0,command=search)
345 searchbut.place(x=1136,y=1043)
346 searchbar.bind("<Return>", lambda event: search())
347
348
349 entryscreen=Frame(win,bg='#b1f8eb')
350 entryscreen.pack(fill=BOTH,expand=True)           # for experiment of login
screen simulation
351 user=Label(entryscreen,image=userpic,bg="#b1f8eb",text='Wellcome User',
352            fg="#004639",font=('lucida handwriting', 20 , 'bold'),compound=TOP)
353 user.place(relx=0.5,rely=0.5,anchor=CENTER)
354
355 signif=Frame(entryscreen,bg='#b1f8eb')
356
357 signin=Entry(signif,fg='gray',bg='#6be4ce',font=('impact',18))
358 signin.insert(0,'Password')
359
360 signin.pack(side=LEFT)

```

```
361
362 signin.bind('<Return>',lambda event: checkpass())
363
364 sbut=Button(signf,fg="#1d5248",activeforeground='#1d5248',bg='#6be4ce',
365             activebackground='#6be4ce',text='ENTER',font=('impact',13),
366             relief='flat',bd=0,highlightthickness=0,command=checkpass)
367 sbut.pack(side=LEFT)
368 signf.place(relx=0.5,rely=0.6,anchor=CENTER)
369 signf.lift()
370 signin.lift()
371
372
373
374
375 win.config(cursor='dotbox')
376 win.bind_all('<Button-1>', global_click_handler, add='+') #call all clicks
377
378
379 bl=Frame(entriyscreen,bg='black')
380 bl.pack(fill=BOTH,expand=True)
381 logo=Label(bl,image=icon,bg='black')
382 logo.place(relx=0.5,rely=0.5,anchor=CENTER)
383 bl.lift()
384 logo.lift()
385 win.update()
386 time.sleep(2)
387 bl.destroy()
388
389
390 win.mainloop()
391 #all icons used are from icon8.com
```