

```

1 from distutils import command
2 from tkinter import *
3 from tkinter import ttk
4 from typing_extensions import AnnotatedAlias
5 from PIL import Image, ImageTk
6 import os
7 import pickle
8 import mysql.connector as sql
9 from tkinter import messagebox
10 from datetime import date
11 from datetime import time
12 from datetime import *
13 import requests
14 from bs4 import BeautifulSoup
15 import time
16 import dashboard
17 import csv
18 import regist_depart
19
20 def depart_screen():
21     def click_delete_department():
22         """when clicked delete departments, it will require to select the departments and after
23         selecting and
24         performing the delete method, it will ask the admin either they are sure they want to
25         delete that department
26         or not if yes then department containing that id in department table is deleted."""
27         try:
28             #obj_department_registration_database = Model_class.department_registration.
29             GetDatabase('use cms;')
30             #db_connection.create(obj_department_registration_database.get_database())
31             a=load_data()
32             host=a[0]
33             username = a[2]
34             password = a[3]
35             port=a[1]
36
37             spec=sql.connect(host=host,user=username,password=password,port=port,database="sms"
38 )
39             mycur=spec.cursor()
40
41             department_view_content = department_tree.focus()
42             department_view_items = department_tree.item(department_view_content)
43             department_view_values = department_view_items['values'][0]
44             ask = messagebox.askyesno("Warning",
45                                     f"Are you sure you want to delete department having id
46 {department_view_values}")
47             if ask is True:
48                 query = f"delete from department where department_id={department_view_values};"
49                 mycur.execute(query)
50                 spec.commit()
51                 #db_connection.delete(query, (department_view_values,))
52                 messagebox.showinfo("Success", f" department id {department_view_values}
53 deleted Successfully")
54
55                 click_view_all()
56             else:
57                 pass
58
59         except BaseException as msg:
60             print(msg)
61             messagebox.showerror("Error",
62                                 "There is some error deleting the data\n Make sure you have
63 Selected the data")

```

```

57
58
59     def click_clear_button():
60         department_code_entry.delete(0, END)
61         department_name_entry.delete(0, END)
62
63
64     def load_data():
65         f=open("Credentials.csv","r")
66         s=csv.reader(f,delimiter="-")
67         d=[]
68         for i in s:
69             d.append(i)
70         a=d[::-1]
71         return (a[0])
72
73     def click_go_to_dashboard():
74         """returns AdminDashboard class when clicked go to dashboard"""
75         dashboard.dashboard()
76         root.withdraw()
77
78     def click_view_all():
79         """it will show all the data contains on the department table of cms database,
80         when clicked by default this method
81         is called while initializing the class ManageDepartment. Exception is handled to
82         avoid run time error which may
83         cause by user."""
84         try:
85             #obj_student_database = Model_class.student_registration.GetDatabase('use cms
86             ;')
87             #db_connection.create(obj_student_database.get_database())
88             a=load_data()
89             host=a[0]
90             username = a[2]
91             password = a[3]
92             port=a[1]
93
94             spec=sql.connect(host=host,user=username,password=password,port=port,database=
95             "sms")
96
97             mycur=spec.cursor()
98             query = "select * from department;"
99             mycur.execute(query)
100
101             data = mycur.fetchall()
102             # print(data)
103             department_tree.delete(*department_tree.get_children())
104             for values in data:
105                 data_list = [values[0], values[1], values[2], values[3]]
106                 # print(data_list)
107                 department_tree.insert('', END, values=data_list)
108
109             except BaseException as msg:
110                 print(msg)
111
112     def updatebutton():
113
114         def tree_event_handle():
115             try:
116                 #obj_department_database = Model_class.department_registration.GetDatabase('
117                 use cms;')
118                 #db_connection.create(obj_department_database.get_database())

```

```

115         tree_view_content = department_tree.focus()
116         tree_view_items = department_tree.item(tree_view_content)
117         # print(tree_view_items)
118         tree_view_values = tree_view_items['values']
119         tree_view_id = tree_view_items['values'][0]
120         list_of_tree.clear()
121         # print(tree_view_id)
122         get_id.clear()
123         get_id.append(tree_view_id)
124         for i in tree_view_values:
125             list_of_tree.append(i)
126
127         ask = messagebox.askyesno("Confirm",
128                                   f"Do you want to Update Student having id {
tree_view_id}")
129         if ask is True:
130             pass
131
132         except BaseException as msg:
133             print(msg)
134             messagebox.showerror("Error",
135                                   "There is some error updating the data\n Make sure you
have Selected the data")
136
137         tree_event_handle()
138
139         def up_dep():
140             try:
141                 #obj_department_database = Model_class.department_registration.GetDatabase('
use cms;')
142                 #db_connection.create(obj_department_database.get_database())
143
144                 #get_id = ManageDepartment.get_id
145                 data_id = get_id[0]
146                 # print(data_id)
147
148                 a=load_data()
149                 host=a[0]
150                 username = a[2]
151                 password = a[3]
152                 port=a[1]
153
154                 spec=sql.connect(host=host,user=username,password=password,port=port,database=
"sms")
155                 mycur=spec.cursor()
156                 #obj_department_database = Model_class.department_registration.
DepartmentRegistration(
157                     #department_code_entry.get(),
158                     #department_name_entry.get(),
159                     #reg_date)
160
161                 query = f"update department set department_code='{department_code_entry.get()}'
',department_name='{department_name_entry.get()}' where department_id={data_id}"
162                 mycur.execute(query)
163                 spec.commit()
164                 click_view_all()
165
166                 #values = (obj_department_database.get_code(), obj_department_database.
get_name(), data_id)
167
168                 #db_connection.update(query, values)
169
170                 ask = messagebox.askyesnocancel("Success",

```

```

171                                     f>Data having \n department Name={
department_code_entry.get()} \n Updated Successfully\n"
172                                     f"Do you want to Go Department Dashboard")
173         if ask is True:
174             pass
175             #win = Toplevel()
176             #Frontend.manage_department.ManageDepartment(win)
177             #window.withdraw()
178             #win.deiconify()
179
180     except BaseException as msg:
181         print(msg)
182         messagebox.showerror("Error", f"Error due to{msg}")
183
184
185
186 a = list_of_tree
187 # print(a)
188
189
190 global department_code_entry, department_name_entry
191
192 root = Toplevel()
193 root.title('DEPARTMENT REGISTRATION FORM - COLLEGE MANAGEMENT SYSTEM')
194 root.geometry('1067x600')
195 root.config(bg="#f29844")
196 root.resizable(False, False)
197
198 # =====Backend connection=====
199 #db_connection = Backend.connection.DatabaseConnection()
200
201 # creating frame for Register
202 # img = img
203 # dummylabel = Label(root, image=img)
204 # dummylabel.place(x=30, y=30)
205
206 reg_frame = Frame(root, bg="#ffffff", width=1000, height=560)
207 reg_frame.place(x=30, y=30)
208
209
210
211
212 heading = Label(reg_frame, text="Department Registration Form", font=('yu gothic ui',
20, "bold"), bg="white",
213                 fg='black',
214                 bd=5,
215                 relief=FLAT)
216 heading.place(x=200, y=0, width=600)
217 #slider()
218 #heading_color()
219
220 dept_frame = LabelFrame(reg_frame, text="Department Details", bg="white", fg="#4f4e4d"
, height=380,
221                        width=800, borderwidth=2.4,
222                        font=("yu gothic ui", 13, "bold"))
223 dept_frame.config(highlightbackground="red")
224 dept_frame.place(x=100, y=90)
225
226 # =====
227 # =====Key Bindings=====
228 # =====
229
230 #root.bind("<Return>", click_enter_submit)

```

```

231
232     # =====
233     # =====Department Code Label
=====
234     # =====
235
236     department_code_label = Label(dept_frame, text="Department Code ", bg="white", fg="#
4f4e4d",
237                                     font=("yu gothic ui", 13, "bold"))
238     department_code_label.place(x=160, y=80)
239
240     department_code_entry = Entry(root, highlightthickness=0, relief=FLAT, bg="white", fg=
"#6b6a69",
241                                     font=("yu gothic ui semibold", 12))
242     department_code_entry.place(x=450, y=227, width=295) # trebuchet ms
243
244     department_code_line = Canvas(root, width=295, height=1.5, bg="#bdb9b1",
highlightthickness=0)
245     department_code_line.place(x=450, y=249)
246
247     # =====
248     # =====Department Name=====
249     # =====
250
251     department_name_label = Label(dept_frame, text="Department Name ", bg="white", fg="#
4f4e4d",
252                                     font=("yu gothic ui", 13, "bold"))
253     department_name_label.place(x=160, y=140)
254
255     department_name_entry = Entry(root, highlightthickness=0, relief=FLAT, bg="white", fg=
"#6b6a69",
256                                     font=("yu gothic ui semibold", 12))
257     department_name_entry.place(x=450, y=287, width=295) # trebuchet ms
258
259     department_name_line = Canvas(root, width=295, height=1.5, bg="#bdb9b1",
highlightthickness=0)
260     department_name_line.place(x=450, y=309)
261
262     reg_date = time.strftime("%Y/%m/%d")
263
264
265
266     # =====
267     # =====Register options=====
268     # =====
269     submit_img = ImageTk.PhotoImage(file='Pics\\submit.png')
270     submit = Button(dept_frame, image=submit_img,
271                     font=("yu gothic ui", 13, "bold"), relief=FLAT,
activebackground="white"
272                     , borderwidth=0, background="white", cursor="hand2",command=
up_dep)
273     submit.image = submit_img
274     submit.place(x=90, y=267)
275
276     clear_img = ImageTk.PhotoImage(file='Pics\\clear.png')
277     clear_button = Button(dept_frame, image=clear_img,
278                           font=("yu gothic ui", 13, "bold"), relief=FLAT,
activebackground="white"
279                           , borderwidth=0, background="white", cursor="hand2",
280                           command=click_clear_button)
281     clear_button.image = clear_img
282     clear_button.place(x=250, y=270)
283

```

```

284     back_img = ImageTk.PhotoImage(file='Pics\\back.png')
285     back_button = Button(dept_frame, image=back_img,
286                          font=("yu gothic ui", 13, "bold"), relief=FLAT,
activebackground="white"
287                          , borderwidth=0, background="white", cursor="hand2")
288     back_button.image = back_img
289     back_button.place(x=410, y=270)
290
291
292     exit_img = ImageTk.PhotoImage(file='Pics\\exit.png')
293     exit_button = Button(dept_frame, image=exit_img,
294                          font=("yu gothic ui", 13, "bold"), relief=FLAT,
activebackground="white"
295                          , borderwidth=0, background="white", cursor="hand2",
command=exit)
296     exit_button.image = exit_img
297     exit_button.place(x=570, y=270)
298
299     try:
300         department_code_entry.insert(0, a[1])
301         department_name_entry.insert(0, a[2])
302
303     except IndexError as msg:
304         print(msg)
305
306
307     list_of_tree = []
308     get_id = []
309
310     root = Toplevel()
311     root.geometry("1067x600")
312     root.title("Manage Department - College Management System")
313     #root.iconbitmap('images\\logo.ico')
314     root.resizable(False, False)
315
316
317     manage_student_frame_r = Image.open('Pics\\student_frame.png').resize((1067,600),Image.
ANTIALIAS)
318     manage_student_frame = ImageTk.PhotoImage(manage_student_frame_r)
319     image_panel = Label(root, image=manage_student_frame)
320     image_panel.image = manage_student_frame
321     image_panel.pack(fill='both', expand='yes')
322
323
324
325     heading = Label(root, text="Department Management Dashboard", font=('yu gothic ui', 20, "
bold"), bg="white",
326                      fg='black',
327                      bd=5,
328                      relief=FLAT)
329     heading.place(x=420, y=23)
330
331
332     # =====
333     # =====Left frame =====
334     # =====
335
336     #left frame
337     left_view_frame = Frame(root, bg="white")
338     left_view_frame.place(x=35, y=89, height=470, width=250)
339
340
341     #tree view frame

```

```

342 tree_view_frame = Frame(root, bg="white")
343 tree_view_frame.place(x=301, y=90, height=473, width=730)
344
345 # =====
346 # =====frame for personal credentials =====
347 # =====
348
349 personal_frame = LabelFrame(left_view_frame, text="Batch Management Options", bg="white",
fg="#4f4e4d",height=460,width=240, borderwidth=2.4,font=("yu gothic ui", 12, "bold"))
350 personal_frame.config(highlightbackground="red")
351 personal_frame.place(x=5, y=8)
352
353
354 # =====
355 # =====Add Department button=====
356 # =====
357
358 add_department_r = Image.open('Pics\\add_department.png').resize((225,25),Image.ANTIALIAS)
359 add_department = ImageTk.PhotoImage(add_department_r)
360 add_department_button = Button(personal_frame, image=add_department, relief=FLAT,
borderwidth=0,
361                                activebackground="white", bg="white", cursor="hand2",
command= regist_depart.regist_depart)
362 add_department_button.image = add_department
363 add_department_button.place(x=8, y=100)
364 # add_student_button.place(x=36, y=295)
365
366 # =====
367 # =====Update employee button=====
368 # =====
369
370 update_department_r = Image.open('Pics\\update_department.png').resize((225,25),Image.
ANTIALIAS)
371 update_department = ImageTk.PhotoImage(update_department_r)
372 update_department_button = Button(personal_frame, image=update_department, relief=FLAT,
borderwidth=0,
373                                activebackground="white", bg="white", cursor="
hand2",command=updatebutton)
374 update_department_button.image = update_department
375 update_department_button.place(x=8, y=165)
376 # update_student_button.place(x=36, y=355)
377
378 # =====
379 # =====Delete employee button=====
380 # =====
381
382 delete_department_r = Image.open('Pics\\delete_department.png').resize((225,25),Image.
ANTIALIAS)
383 delete_department = ImageTk.PhotoImage(delete_department_r)
384 delete_department_button = Button(personal_frame, image=delete_department, relief=FLAT,
borderwidth=0,
385                                activebackground="white", bg="white", cursor="
hand2",command=click_delete_department)
386 delete_department_button.image = delete_department
387 delete_department_button.place(x=8, y=230)
388 # delete_student_button.place(x=36, y=405)
389
390 # =====
391 # =====Goto Main dashboard button=====
392 # =====
393
394 goto_dashboard_r = Image.open('Pics\\goto_dashboard.png').resize((225,25),Image.ANTIALIAS)
395 goto_dashboard = ImageTk.PhotoImage (goto_dashboard_r)

```

```

396     goto_dashboard_button = Button(personal_frame, image=goto_dashboard, relief=FLAT,
borderwidth=0,activebackground="white", bg="white", cursor="hand2",command=
click_go_to_dashboard)
397     goto_dashboard_button.image = goto_dashboard
398     goto_dashboard_button.place(x=5, y=295)
399
400     # =====
401     # =====Starting Tree View=====
402     # =====
403     style = ttk.Style()
404     style.configure("Treeview.Heading", font=('yu gothic ui', 10, "bold"), foreground="red")
405
406     scroll_x = Scrollbar(tree_view_frame, orient=HORIZONTAL)
407     scroll_y = Scrollbar(tree_view_frame, orient=VERTICAL)
408     department_tree = ttk.Treeview(tree_view_frame,
409                                   columns=(
410                                       "DEPARTMENT ID", "DEPARTMENT CODE", "DEPARTMENT
NAME", "REGISTRATION DATE"),
411                                   xscrollcommand=scroll_x.set, yscrollcommand=scroll_y.
set)
412     scroll_x.pack(side=BOTTOM, fill=X)
413     scroll_y.pack(side=RIGHT, fill=Y)
414     scroll_x.config(command=department_tree.xview)
415     scroll_y.config(command=department_tree.yview)
416
417     # =====TreeView Heading=====
418     department_tree.heading("DEPARTMENT ID", text="DEPARTMENT ID")
419     department_tree.heading("DEPARTMENT CODE", text="DEPARTMENT CODE")
420     department_tree.heading("DEPARTMENT NAME", text="DEPARTMENT NAME")
421     department_tree.heading("REGISTRATION DATE", text="REGISTRATION DATE")
422     department_tree["show"] = "headings"
423
424     # =====TreeView Column=====
425     department_tree.column("DEPARTMENT ID", width=100)
426     department_tree.column("DEPARTMENT CODE", width=100)
427     department_tree.column("DEPARTMENT NAME", width=170)
428     department_tree.column("REGISTRATION DATE", width=100)
429     department_tree.pack(fill=BOTH, expand=1)
430
431     #department_tree.bind("<Delete>", click_delete_key)
432     #department_tree.bind("<Button-3>", do_popup)
433     #department_tree.bind("<Double-Button-1>", tree_double_click)
434     #department_tree.bind("<Return>", tree_double_click)
435     #search_start()
436     #search_by.current(0)
437     click_view_all()
438
439     #root.mainloop()
440
441
442 if __name__ == "__main__":
443     depart_screen()

```