```python
from distutils import command
from tkinter import *
from tkinter import ttk
from turtle import up
from typing_extensions import _AnnotatedAlias
from PIL import Image,ImageTk
import os
import pickle
import mysql.connector  as sql
from tkinter import messagebox
from datetime import date
from datetime import time
from datetime import *
import requests
from bs4 import BeautifulSoup
import time
import dashboard
import csv
import regist_section


def section_screen():
    def load_data():
        f=open("Credentials.csv","r")
        s=csv.reader(f,delimiter="-")
        d=[]
        for i in s:
            d.append(i)
        a=d[::-1]
        return (a[0])

    def click_go_to_dashboard():
        """returns AdminDashboard class when clicked go to dashboard"""
        dashboard.dashboard()
        root.withdraw()

    def click_view_all():
            """it will show all the data contains on the section table of cms database, when clicked by default this method
            is called while initializing the class ManageSection. Exception is handled to avoid run time error which may
            cause by user."""
            try:
                #obj_student_database = Model_class.student_registration.GetDatabase('use cms;')
                #db_connection.create(obj_student_database.get_database())
                a=load_data()
                host=a[0]
                username = a[2]
                password = a[3]
                port=a[1]

                spec=sql.connect(host=host,user=username,password=password,port=port,database="sms")
                mycur=spec.cursor()
                query = "select * from section;"
                mycur.execute(query)
                data = mycur.fetchall()
                # print(data)
                section_tree.delete(*section_tree.get_children())
                for values in data:
                    data_list = [values[0], values[1], values[2], values[3], values[4]]
                    # print(data_list)
```

```
60                      section_tree.insert('', END, values=data_list)
61
62              except BaseException as msg:
63                  print(msg)
64
65
66      def click_delete_section():
67              """when clicked delete sections, it will require to select the sections and after selecting and
68              performing the delete method, it will ask the admin either they are sure they want to delete that section
69              or not if yes then section containing that id in section table is deleted."""
70              try:
71                  #obj_section_registration_database = Model_class.section_registration.GetDatabase('use cms;')
72                  #self.db_connection.create(obj_section_registration_database.get_database())
73                  a=load_data()
74                  host=a[0]
75                  username = a[2]
76                  password = a[3]
77                  port=a[1]
78
79                  spec=sql.connect(host=host,user=username,password=password,port=port,database="sms")
80                  mycur=spec.cursor()
81
82                  section_view_content = section_tree.focus()
83                  section_view_items = section_tree.item(section_view_content)
84                  section_view_values = section_view_items['values'][0]
85                  ask = messagebox.askyesno("Warning",
86                                            f"Are you sure you want to delete Section having id {section_view_values}")
87                  if ask is True:
88                      query = f"delete from section where section_id={section_view_values};"
89                      mycur.execute(query)
90                      spec.commit()
91                      #self.db_connection.delete(query, (section_view_values,))
92                      messagebox.showinfo("Success", f" Section id {section_view_values} deleted Successfully")
93
94                      click_view_all()
95                  else:
96                      print("not deleted")
97
98              except BaseException as msg:
99                  print(msg)
100                 messagebox.showerror("Error",
101                                      "There is some error deleting the data\n Make sure you have Selected the data")
102
103     def up_sec():
104
105
106         def tree_event_handle():
107             try:
108                 #obj_student_database = Model_class.student_registration.GetDatabase('use cms;')
109                 #self.db_connection.create(obj_student_database.get_database())
110                 a=load_data()
111                 host=a[0]
112                 username = a[2]
113                 password = a[3]
114                 port=a[1]
```

```
115
116                 spec=sql.connect(host=host,user=username,password=password,port=port,database=
    "sms")
117                 mycur=spec.cursor()
118
119                 tree_view_content = section_tree.focus()
120                 tree_view_items = section_tree.item(tree_view_content)
121                 # print(tree_view_items)
122                 tree_view_values = tree_view_items['values']
123                 tree_view_id = tree_view_items['values'][0]
124                 # print(tree_view_id)
125                 get_id.clear()
126                 list_of_tree.clear()
127                 get_id.append(tree_view_id)
128                 for i in tree_view_values:
129                     list_of_tree.append(i)
130
131                 ask = messagebox.askyesno("Confirm",
132                                           f"Do you want to Update Student having id {
    tree_view_id}")
133                 if ask is True:
134                     pass
135
136             except BaseException as msg:
137                 print(msg)
138                 messagebox.showerror("Error",
139                                      "There is some error updating the data\n Make sure you
    have Selected the data")
140         #Frontend.section_registration.Clock(wind)
141
142         tree_event_handle()
143         a = list_of_tree
144         # print(a)
145
146         '''
147         if count <= len(txt):
148             count = 0
149             text = ''
150             heading.config(text=text)
151         heading.place(x=150, y=0, width=800)'''
152
153         def update():
154             """updates the data of section from entry fields"""
155             try:
156                 #obj_section_database = Model_class.section_registration.GetDatabase('use cms
    ;')
157                 #db_connection.create(obj_section_database.get_database())
158
159                 #get_id = get_id
160                 data_id = get_id[0]
161                 # print(data_id)
162                 a=load_data()
163                 host=a[0]
164                 username = a[2]
165                 password = a[3]
166                 port=a[1]
167
168                 spec=sql.connect(host=host,user=username,password=password,port=port,database=
    "sms")
169                 mycur=spec.cursor()
170                 #obj_section_database = Model_class.section_registration.SectionRegistration(
    section_code_entry.get(),
171                                                                                          #
```

```
171 section_name_entry.get(),
172                                                                                 #
    section_capacity_entry.get(),
173                                                                                 #
    reg_date)
174                 query = f"update section set section_code='{section_code_entry.get()}',
    section_name='{section_name_entry.get()}', section_capacity='{section_capacity_entry.get()}'"
     \
175                         f" where section_id='{data_id}'"
176                 mycur.execute(query)
177                 spec.commit()
178
179
180                 #values = (obj_section_database.get_code(), obj_section_database.get_name(),
181                             #obj_section_database.get_capacity(), data_id)
182
183                 #db_connection.update(query, values)
184
185                 ask = messagebox.askyesnocancel("Success", f"Data having \n Section Name={
    section_name_entry.get()} \n Updated Successfully\n"
186                                                             f"Do you want to Go Section
    Dashboard")
187
188                 click_view_all()
189                 if ask is True:
190                     pass
191                     #win = Toplevel()
192                     #Frontend.manage_section.ManageSection(win)
193                     #window.withdraw()
194                     #win.deiconify()
195
196             except BaseException as msg:
197                 print(msg)
198                 messagebox.showerror("Error", f"Error due to{msg}")
199
200         global section_code_entry,section_name_entry,section_capacity_entry
201
202         def click_clear_button():
203                 section_name_entry.delete(0, END)
204                 section_code_entry.delete(0, END)
205                 section_capacity_entry.delete(0,END)
206
207         root = Toplevel()
208         root.title('SECTION REGISTRATION FORM - COLLEGE MANAGEMENT SYSTEM')
209         root.geometry('1067x600')
210         root.config(bg="#f29844")
211
212         # ======================Backend connection=============
213         #db_connection = Backend.connection.DatabaseConnection()
214
215         reg_frame = Frame(root, bg="#ffffff", width=1000, height=560)
216         reg_frame.place(x=30, y=30)
217
218
219
220         heading = Label(reg_frame, text="Section Registration Form", font=('yu gothic ui', 30
    , "bold"), bg="white",
221                                 fg='black',
222                                 bd=5,
223                                 relief=FLAT)
224         heading.place(x=200, y=0, width=600)
225
226
```

```
227         section_frame = LabelFrame(reg_frame, text="Section Details", bg="white", fg="#4f4e4d"
    , height=380,
228                                    width=800, borderwidth=2.4,
229                                    font=("yu gothic ui", 13, "bold"))
230         section_frame.config(highlightbackground="red")
231         section_frame.place(x=100, y=90)
232 
233         # ========================================================================
234         # ============================Key Bindings====================================
235         # ========================================================================
236 
237         #root.bind("<Return>", click_enter_submit)
238 
239         # # ========================================================================
240         # # ============================Section ID label====================================
241         # # ========================================================================
242 
243         # ========================================================================
244         # ============================Subject Code ====================================
245         # ========================================================================
246 
247         section_code_label = Label(section_frame, text="Section Code ", bg="white", fg="#
    4f4e4d",
248                                    font=("yu gothic ui", 13, "bold"))
249         section_code_label.place(x=160, y=50)
250 
251         section_code_entry = Entry(root, highlightthickness=0, relief=FLAT, bg="white", fg="#
    6b6a69",
252                                    font=("yu gothic ui semibold", 12))
253         section_code_entry.place(x=405, y=197, width=340)  # trebuchet ms
254 
255         section_code_line = Canvas(root, width=340, height=1.5, bg="#bdb9b1",
    highlightthickness=0)
256         section_code_line.place(x=405, y=219)
257 
258         # ========================================================================
259         # ============================Section Name====================================
260         # ========================================================================
261 
262         section_name_label = Label(section_frame, text="Section Name ", bg="white", fg="#
    4f4e4d",
263                                    font=("yu gothic ui", 13, "bold"))
264         section_name_label.place(x=160, y=100)
265 
266         section_name_entry = Entry(root, highlightthickness=0, relief=FLAT, bg="white", fg="#
    6b6a69",
267                                    font=("yu gothic ui semibold", 12))
268         section_name_entry.place(x=410, y=247, width=335)  # trebuchet ms
269 
270         section_name_line = Canvas(root, width=335, height=1.5, bg="#bdb9b1",
    highlightthickness=0)
271         section_name_line.place(x=410, y=269)
272 
273         # ========================================================================
274         # ============================Section capacity====================================
275         # ========================================================================
276         root.option_add("*TCombobox*Listbox*Foreground", '#f29844')
277 
278         section_capacity_label = Label(section_frame, text="Section Capacity ", bg="white", fg
    ="#4f4e4d",
279                                        font=("yu gothic ui", 13, "bold"))
280         section_capacity_label.place(x=160, y=150)
281 
```

```
282         section_capacity_entry = Entry(root, highlightthickness=0, relief=FLAT, bg="white", fg
    ="#6b6a69",
283                                        font=("yu gothic ui semibold", 12))
284         section_capacity_entry.place(x=430, y=297, width=315)  # trebuchet ms
285 
286         section_capacity_line = Canvas(root, width=315, height=1.5, bg="#bdb9b1",
    highlightthickness=0)
287         section_capacity_line.place(x=430, y=319)
288 
289         reg_date = time.strftime("%Y/%m/%d")
290 
291 
292 
293         # ========================================================================
294         # ============================Register options=====================================
295         # ========================================================================
296         submit_img = ImageTk.PhotoImage(file='Pics\\submit.png')
297         submit = Button(section_frame, image=submit_img,
298                                 font=("yu gothic ui", 13, "bold"), relief=FLAT,
    activebackground="white"
299                                 , borderwidth=0, background="white", cursor="hand2",command=
    update)
300         submit.image = submit_img
301         submit.place(x=90, y=267)
302 
303 
304         clear_img = ImageTk.PhotoImage(file='Pics\\clear.png')
305         clear_button = Button(section_frame, image=clear_img,
306                                     font=("yu gothic ui", 13, "bold"), relief=FLAT,
    activebackground="white"
307                                     , borderwidth=0, background="white", cursor="hand2",
308                                     command=click_clear_button)
309         clear_button.image = clear_img
310         clear_button.place(x=250, y=270)
311 
312 
313         back_img = ImageTk.PhotoImage(file='Pics\\back.png')
314         back_button = Button(section_frame, image=back_img,
315                                     font=("yu gothic ui", 13, "bold"), relief=FLAT,
    activebackground="white"
316                                     , borderwidth=0, background="white", cursor="hand2")
317         back_button.image = back_img
318         back_button.place(x=410, y=270)
319 
320 
321         exit_img = ImageTk.PhotoImage(file='Pics\\exit.png')
322         exit_button = Button(section_frame, image=exit_img,
323                                     font=("yu gothic ui", 13, "bold"), relief=FLAT,
    activebackground="white"
324                                     , borderwidth=0, background="white", cursor="hand2",
    command=exit)
325         exit_button.image = exit_img
326         exit_button.place(x=570, y=270)
327 
328         try:
329             section_code_entry.insert(0, a[1])
330             section_name_entry.insert(0, a[2])
331             section_capacity_entry.insert(0, a[3])
332             submit.configure(command=update)
333             txt = f" You are updating Section '{a[2]}'"
334         except IndexError as msg:
335             print(msg)
336 
```

```
337     list_of_tree = []
338     get_id = []
339 
340     root = Toplevel()
341     root.geometry("1067x600")
342     root.title("Section Management Dashboard - College Management System")
343     #root.iconbitmap('images\\logo.ico')
344     root.resizable(False, False)
345 
346     manage_student_frame_r = Image.open('Pics\\student_frame.png').resize((1067,600),Image.
    ANTIALIAS)
347     manage_student_frame = ImageTk.PhotoImage(manage_student_frame_r)
348     image_panel = Label(root, image=manage_student_frame)
349     image_panel.image = manage_student_frame
350     image_panel.pack(fill='both', expand='yes')
351 
352     #b_connection = Backend.connection.DatabaseConnection()
353 
354     txt = "Section Management Dashboard"
355     count = 0
356     text = ''
357     color = ["#4f4e4d", "#f29844", "#fa3939"]
358     heading = Label(root, text=txt, font=('yu gothic ui', 20, "bold"), bg="white",
359                             fg='black',
360                             bd=5,
361                             relief=FLAT)
362     heading.place(x=420, y=26, width=640)
363     #slider()
364     #heading_color()
365 
366     # ============================================================================
367     # ========================Defining Variables==================================
368     # ============================================================================
369 
370     #search_by_var = StringVar()
371     #sort_by_var = StringVar()
372     #sort_in_var = StringVar()
373     #sort_date_in_var = StringVar()
374 
375     # ============================================================================
376     #left frame
377     left_view_frame = Frame(root, bg="white")
378     left_view_frame.place(x=35, y=89, height=470, width=250)
379 
380 
381     #tree view frame
382     tree_view_frame = Frame(root, bg="white")
383     tree_view_frame.place(x=301, y=90, height=473, width=730)
384 
385     # ============================================================================
386     # ========================frame for personal credentials =================
387     # ============================================================================
388 
389     personal_frame = LabelFrame(left_view_frame, text="Section Management Options", bg="white"
    , fg="#4f4e4d",height=460,width=240, borderwidth=2.4,font=("yu gothic ui", 12, "bold"))
390     personal_frame.config(highlightbackground="red")
391     personal_frame.place(x=5, y=8)
392 
393     # ============================================================================
394     # ============================Add Subject button==============================
395     # ============================================================================
396     add_section_r = Image.open("Pics\\add_section.png").resize((225,25),Image.ANTIALIAS)
397     add_section = ImageTk.PhotoImage(add_section_r)
```

```
398     add_section_button = Button(personal_frame, image=add_section, relief=FLAT, borderwidth=0,
    activebackground="white", bg="white", cursor="hand2",command=regist_section.sec_reg)
399     add_section_button.image = add_section
400     add_section_button.place(x=5, y=100)
401     # add_student_button.place(x=36, y=295)
402 
403     # ============================================================================
404     # ============================Update subject button=============================
405     # ============================================================================
406 
407     update_section_r = Image.open('Pics\\update_section.png').resize((225,25),Image.ANTIALIAS)
408     update_section = ImageTk.PhotoImage(update_section_r)
409     update_section_button = Button(personal_frame, image=update_section, relief=FLAT,
    borderwidth=0,
410                                    activebackground="white", bg="white", cursor="hand2",
    command=up_sec
411                                    )
412     update_section_button.image = update_section
413     update_section_button.place(x=5, y=165)
414     # update_student_button.place(x=36, y=355)
415 
416     # ============================================================================
417     # ============================Delete section button=============================
418     # ============================================================================
419 
420     delete_section_r = Image.open('Pics\\delete_section.png').resize((225,25),Image.ANTIALIAS)
421     delete_section = ImageTk.PhotoImage(delete_section_r)
422     delete_section_button = Button(personal_frame, image=delete_section, relief=FLAT,
    borderwidth=0,
423                                    activebackground="white", bg="white", cursor="hand2",
    command=click_delete_section
424                                    )
425     delete_section_button.image = delete_section
426     delete_section_button.place(x=5, y=230)
427     # delete_student_button.place(x=36, y=405)
428 
429     # ============================================================================
430     # ============================Goto Main dashboard button=============================
431     # ============================================================================
432 
433     goto_dashboard_r = Image.open('Pics\\goto_dashboard.png').resize((225,25),Image.ANTIALIAS)
434     goto_dashboard = ImageTk.PhotoImage (goto_dashboard_r)
435     goto_dashboard_button = Button(personal_frame, image=goto_dashboard, relief=FLAT,
    borderwidth=0,activebackground="white", bg="white", cursor="hand2",command=
    click_go_to_dashboard)
436     goto_dashboard_button.image = goto_dashboard
437     goto_dashboard_button.place(x=5, y=295)
438 
439     # ============================================================================
440     # ========================Starting Tree View============================
441     # ============================================================================
442     style = ttk.Style()
443     style.configure("Treeview.Heading", font=('yu gothic ui', 10, "bold"), foreground="red")
444 
445     scroll_x = Scrollbar(tree_view_frame, orient=HORIZONTAL)
446     scroll_y = Scrollbar(tree_view_frame, orient=VERTICAL)
447     section_tree = ttk.Treeview(tree_view_frame,
448                                 columns=(
449                                     "SECTION ID", "SECTION CODE", "SECTION NAME", "
    SECTION CAPACITY",
450                                     "REGISTRATION DATE"),
451                                 xscrollcommand=scroll_x.set, yscrollcommand=scroll_y.
    set)
```

```
452     scroll_x.pack(side=BOTTOM, fill=X)
453     scroll_y.pack(side=RIGHT, fill=Y)
454     scroll_x.config(command=section_tree.xview)
455     scroll_y.config(command=section_tree.yview)
456 
457     # ============================TreeView Heading====================
458     section_tree.heading("SECTION ID", text="SECTION ID")
459     section_tree.heading("SECTION CODE", text="SECTION CODE")
460     section_tree.heading("SECTION NAME", text="SECTION NAME")
461     section_tree.heading("SECTION CAPACITY", text="SECTION CAPACITY")
462     section_tree.heading("REGISTRATION DATE", text="REGISTRATION DATE")
463     section_tree["show"] = "headings"
464 
465     # ============================TreeView Column====================
466     section_tree.column("SECTION ID", width=50)
467     section_tree.column("SECTION CODE", width=100)
468     section_tree.column("SECTION NAME", width=150)
469     section_tree.column("SECTION CAPACITY", width=100)
470     section_tree.column("REGISTRATION DATE", width=100)
471     section_tree.pack(fill=BOTH, expand=1)
472 
473     #section_tree.bind("<Delete>", click_delete_key)
474     #section_tree.bind("<Button-3>", do_popup)
475     #section_tree.bind("<Double-Button-1>", tree_double_click)
476     #section_tree.bind("<Return>", tree_double_click)
477     #search_start()
478     #search_by.current(0)
479     click_view_all()
480 
481     #root.mainloop()
482 
483 if __name__ =="__main__":
484     section_screen()
```