

```

1 from cProfile import label
2 from distutils import command
3 from logging import root
4 from tkinter import *
5 from tkinter import ttk
6 from PIL import Image,ImageTk
7 import os
8 import pickle
9 import mysql.connector as sql
10 from tkinter import messagebox
11 from datetime import date
12 from datetime import time
13 from datetime import *
14 import requests
15 from bs4 import BeautifulSoup
16 import time
17 import user_inter
18 import csv
19 import course_screen
20 import section_screen
21 import batch_screen
22 import department_screen
23 import emp_screen
24 import student_screen
25
26
27 def load_data():
28     f=open("Credentials.csv","r")
29     s=csv.reader(f,delimiter="-")
30     d=[]
31     for i in s:
32         d.append(i)
33     a=d[::-1]
34     return (a[0])
35
36 def click_logout():
37     root.destroy()
38
39 def dashboard():
40
41     headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.3'}
42
43     def click_home():
44         """set to default home tab where details like no. of students, employees, department shows """
45
46         home_frame = Frame(root)
47         home_frame.place(x=120, y=105, height=440, width=910)
48
49         home_dashboard_fram_r=Image.open('files\home_new.png')
50         home_dashboard_fram_r=home_dashboard_fram_r.resize((910,440),Image.ANTIALIAS)
51         home_dashboard_frame = ImageTk.PhotoImage(home_dashboard_fram_r)
52         home_panel = Label(home_frame, image=home_dashboard_frame, bg="white")
53         home_panel.image= home_dashboard_frame
54         home_panel.pack(fill='both', expand='yes')
55
56     try:
57         #obj_student_database = Model_class.student_registration.GetDatabase('use cms;')
58         #self.db_connection.create(obj_student_database.get_database())
59         a=load_data()
60         host=a[0]
61         username = a[2]

```

```

62         password = a[3]
63         port=a[1]
64
65         print(a)
66         spec=sql.connect(host=host,user=username,password=password,port=port,database="sms"
")
67         mycur=spec.cursor()
68
69         query = "SELECT COUNT(*) FROM students;"
70         mycur.execute(query)
71         data=mycur.fetchall()
72
73
74
75         global no_students
76         for value in data:
77             no_students = value[0]
78
79         total_students = Label(home_frame, text=f" TOTAL\n {no_students}\n STUDENTS",
80                             font=("yu gothic ui", 20, "bold"),
81                             background="white", fg="#e67c0b")
82         total_students.place(x=180, y=65)
83
84         query = "SELECT COUNT(*) FROM employees;"
85         mycur.execute(query)
86         data = mycur.fetchall()
87         global no_employees
88         for value in data:
89             no_employees = value[0]
90
91         total_employees = Label(home_frame, text=f" TOTAL\n {no_employees}\n EMPLOYEES",
92                             font=("yu gothic ui", 20, "bold"),
93                             background="white", fg="#e67c0b")
94         total_employees.place(x=550, y=65)
95
96         query = "SELECT COUNT(*) FROM department;"
97         mycur.execute(query)
98         data = mycur.fetchall()
99         global no_department
100        for value in data:
101            no_department = value[0]
102
103        total_department = Label(home_frame, text=f" TOTAL\n {no_department}\n DEPARTMENTS
",
104                             font=("yu gothic ui", 20, "bold"),
105                             background="white", fg="#e67c0b")
106        total_department.place(x=175, y=250)
107
108    except BaseException as msg:
109        print(msg)
110
111    def click_manage():
112        """ opens new frame from where one can go to manage students, employees,
113        departments, course, section
114        and batch"""
115
115        manage_frame = Frame(root, bg="white")
116        manage_frame.place(x=120, y=105, height=440, width=910)
117        manage_dashboard_frame_r = Image.open('files\\manage_frame1.png')
118        manage_dashboard_frame_r=manage_dashboard_frame_r.resize((910,440),Image.ANTIALIAS
)
119        manage_dashboard_frame = ImageTk.PhotoImage(manage_dashboard_frame_r)
120        manage_panel = Label(manage_frame, image=manage_dashboard_frame, bg="white")

```

```

121     manage_panel.image = manage_dashboard_frame
122     manage_panel.pack(fill='both', expand='yes')
123
124     student_r=Image.open('Pics\\student.png')
125     student_r=student_r.resize((100,100),Image.ANTIALIAS)
126     student = ImageTk.PhotoImage(student_r)
127     student_button = Button(manage_frame, image=student, relief=FLAT, borderwidth=0,
128     activebackground="white", bg="white", cursor="hand2", command=student_screen.stu_screen)
129         student_button.image = student
130         student_button.place(x=120, y=115)
131
131     employee_r=Image.open('Pics\\employee.png')
132     employee_r=employee_r.resize((100,100),Image.ANTIALIAS)
133     employee = ImageTk.PhotoImage(employee_r)
134     employee_button = Button(manage_frame, image=employee, relief=FLAT, borderwidth=0,
135     activebackground="white", bg="white", cursor="hand2", command=emp_screen.emp_screen)
136         employee_button.image = employee
136         employee_button.place(x=395, y=115)
137
138     department_r=Image.open('Pics\\department.png')
139     department_r=department_r.resize((100,100),Image.ANTIALIAS)
140     department = ImageTk.PhotoImage(department_r)
141     department_button = Button(manage_frame, image=department, relief=FLAT,
142     borderwidth=0,activebackground="white", bg="white", cursor="hand2", command=department_screen.
142     depart_screen)
142         department_button.image = department
143         department_button.place(x=672, y=115)
144
145     course_r=Image.open('Pics\\course.png')
146     course_r=course_r.resize((100,100),Image.ANTIALIAS)
147     course = ImageTk.PhotoImage(course_r)
148     course_button = Button(manage_frame, image=course, relief=FLAT, borderwidth=0,
149     activebackground="white", bg="white", cursor="hand2", command=course_screen.course_screen)
149         course_button.image = course
150         course_button.place(x=121, y=300)
151
152     section_r=Image.open('Pics\\section.png')
153     section_r=section_r.resize((100,100),Image.ANTIALIAS)
154     section = ImageTk.PhotoImage(section_r)
155     section_button = Button(manage_frame, image=section, relief=FLAT, borderwidth=0,
156     activebackground="white", bg="white", cursor="hand2", command=section_screen.section_screen)
156         section_button.image = section
157         section_button.place(x=396, y=300)
158
159
160     batch_r=Image.open('Pics\\batch.png')
161     batch_r=batch_r.resize((100,100),Image.ANTIALIAS)
162     batch = ImageTk.PhotoImage(batch_r)
163     batch_button = Button(manage_frame, image=batch, relief=FLAT, borderwidth=0,
164     activebackground="white", bg="white", cursor="hand2", command= batch_screen.batch_screen)
164         batch_button.image = batch
165         batch_button.place(x=673, y=300)
166
167     def click_view():
168         """ Displays partial data into tree view of students, employees, departments, courses
168         when clicked view tab
169             on interface """
170         view_frame = Frame(root, bg="white")
171         view_frame.place(x=120, y=105, height=440, width=910)
172
173         view_dashboard_frame_r=Image.open('Pics\\view_frame.png')
174         view_dashboard_frame_r=view_dashboard_frame_r.resize((910,440),Image.ANTIALIAS)
175         view_dashboard_frame = ImageTk.PhotoImage(view_dashboard_frame_r)

```

```

176     view_panel = Label(view_frame, image=view_dashboard_frame, bg="white")
177     view_panel.image=view_dashboard_frame
178     view_panel.pack(fill='both', expand='yes')
179
180
181
182
183     #department_view_label = Label(view_frame, text="View Department Information ", bg="white", fg="#4f4e4d",font=("yu gothic ui", 13, "bold"))
184     #department_view_label.place(x=770, y=290)
185
186     #course_view_label = Button(view_frame, text="View Course Information ", bg="white", fg="#4f4e4d",font=("yu gothic ui", 13, "bold"),command=studentinfo)
187     #course_view_label.place(x=170, y=290)
188
189
190
191     def studentinfo():
192         # =====
193         # =====Displaying Student Information=====
194         # =====
195         def view_student_information():
196             """fetched data of students from database and inserted required index to student tree view"""
197
198             try:
199                 obj_student_database = Model_class.student_registration.GetDatabase('use cms; ')
200                 self.db_connection.create(obj_student_database.get_database())
201                 a=load_data()
202                 host=a[0]
203                 username = a[2]
204                 password = a[3]
205                 port=a[1]
206
207                 spec=sql.connect(host=host,user=username,password=password,port=port,
208 database="sms"))
209                 mycur=spec.cursor()
210
211                 query = "select * from students;"
212                 mycur.execute(query)
213                 data = mycur.fetchall()
214                 self.a = Label(self.window)
215                 # print(data)
216                 self.view_student_tree.delete(*self.view_student_tree.get_children())
217                 for values in data:
218                     data_list = [values[0], values[4], values[11], values[9]]
219                     # print(data_list)
220                     view_student_tree.insert('', END, values=data_list)
221
222             except BaseException as msg:
223                 print(msg)
224
225                 student_view_label = Label(view_frame, text="View Students Information ", bg="white", fg="#4f4e4d",font=("yu gothic ui", 13, "bold"))
226                 student_view_label.place(x=170, y=6)
227
228                 view_student_frame = Toplevel()#Frame(view_frame, bg="white")
229                 view_student_frame.geometry("800x500")
230                 #view_student_frame.place(x=10, y=40, height=250, width=575)
231
232                 style = ttk.Style()
233                 style.configure("Treeview.Heading", font=('yu gothic ui', 10, "bold"), foreground=

```

```

232 "red")
233         #style.configure("Treeview", font=('yu gothic ui', 9, "bold"), foreground="black")
234
235         scroll_y = Scrollbar(view_student_frame, orient=VERTICAL)
236         scroll_x = Scrollbar(view_student_frame, orient=HORIZONTAL)
237         view_student_tree = ttk.Treeview(view_student_frame,columns=("STUDENT ID", "
238             STUDENT NAME", "COURSE ENROLLED", "PHONE NO."),xscrollcommand=scroll_x.set, yscrollcommand=
239             scroll_y.set,height=2)
240             scroll_x.pack(side=BOTTOM, fill=X)
241             scroll_y.pack(side=RIGHT, fill=Y)
242             scroll_x.config(command=view_student_tree.xview)
243             scroll_y.config(command=view_student_tree.yview)
244
245             # =====TreeView Heading=====
246             view_student_tree.heading("STUDENT ID", text="STUDENT ID")
247             view_student_tree.heading("STUDENT NAME", text="STUDENT NAME")
248             view_student_tree.heading("COURSE ENROLLED", text="COURSE ENROLLED")
249             view_student_tree.heading("PHONE NO.", text="PHONE NO.")
250             view_student_tree["show"] = "headings"
251
252             # =====TreeView Column=====
253             view_student_tree.column("STUDENT ID", width=50)
254             view_student_tree.column("STUDENT NAME", width=150)
255             view_student_tree.column("COURSE ENROLLED", width=100)
256             view_student_tree.column("PHONE NO.", width=100)
257             view_student_tree.pack(fill=BOTH, expand=1)
258
259             view_student_information()
260             #course_view_label = Button(view_frame, text="View Course Information ", bg="white",
261             fg="#4f4e4d",font=("yu gothic ui", 13, "bold"),command=studentinfo)
262             #course_view_label.place(x=170, y=290)
263
264             def employee_info():
265                 # =====Displaying instructor Information=====
266
267                 def view_employee_information():
268                     """fetched data of employees from database and inserted required index to
269                     employee tree view"""
270                     try:
271                         obj_student_database = Model_class.student_registration.GetDatabase('use
272                         cms;')
273                         #self.db_connection.create(obj_student_database.get_database())
274                         a=load_data()
275                         host=a[0]
276                         username = a[2]
277                         password = a[3]
278                         port=a[1]
279
280                         spec=sql.connect(host=host,user=username,password=password,port=port,
281                         database="sms")
282                         mycur=spec.cursor()
283
284                         query = "select * from employees;"
285                         mycur.execute(query)
286                         data = mycur.fetchall()
287                         #self.a = Label(self.window)
288                         # print(data)
289                         #self.view_student_tree.delete(*self.view_student_tree.get_children())
290                         for values in data:
291                             data_list = [values[0], values[4], values[13], values[11]]

```

```

289         # print(data_list)
290         view_employee_tree.insert('', END, values=data_list)
291
292     except BaseException as msg:
293         print(msg)
294
295     view_employee_frame = Toplevel()#(view_frame, bg="white")
296     view_employee_frame.geometry("800x500")
297     #view_employee_frame.place(x=595, y=40, height=250, width=575)
298     #employee_view_label = Label(view_frame, text="View Employees Information ", bg="white", fg="#4f4e4d",font=("yu gothic ui", 13, "bold"))
299     #employee_view_label.place(x=600, y=5)
300
301     scroll_y_e = Scrollbar(view_employee_frame, orient=VERTICAL)
302     scroll_x_e = Scrollbar(view_employee_frame, orient=HORIZONTAL)
303     view_employee_tree = ttk.Treeview(view_employee_frame,columns=("EMPLOYEE ID", "EMPLOYEE NAME", "DEPARTMENT", "EMPLOYEE TYPE"),xscrollcommand=scroll_x_e.set, yscrollcommand=scroll_y_e.set)
304     scroll_x_e.pack(side=BOTTOM, fill=X)
305     scroll_y_e.pack(side=RIGHT, fill=Y)
306     scroll_x_e.config(command=view_employee_tree.xview)
307     scroll_y_e.config(command=view_employee_tree.yview)
308
309     # =====TreeView Heading=====
310     view_employee_tree.heading("EMPLOYEE ID", text="EMPLOYEE ID")
311     view_employee_tree.heading("EMPLOYEE NAME", text="EMPLOYEE NAME")
312     view_employee_tree.heading("DEPARTMENT", text="DEPARTMENT")
313     view_employee_tree.heading("EMPLOYEE TYPE", text="EMPLOYEE TYPE")
314     view_employee_tree["show"] = "headings"
315
316     # =====TreeView Column=====
317     view_employee_tree.column("EMPLOYEE ID", width=50)
318     view_employee_tree.column("EMPLOYEE NAME", width=150)
319     view_employee_tree.column("DEPARTMENT", width=100)
320     view_employee_tree.column("EMPLOYEE TYPE", width=100)
321     view_employee_tree.pack(fill=BOTH, expand=1)
322
323     view_employee_information()
324
325     def course_info():
326         # =====
327         # =====Displaying Course Information=====
328         # =====
329         def view_course_information():
330             """fetched data of course from database and inserted required index to course
tree view"""
331             try:
332                 obj_student_database = Model_class.student_registration.GetDatabase('use
cms; ')
333                 #self.db_connection.create(obj_student_database.get_database())
334                 a=load_data()
335                 host=a[0]
336                 username = a[2]
337                 password = a[3]
338                 port=a[1]
339
340                 spec=sql.connect(host=host,user=username,password=password,port=port,
341 database="sms")
342                 mycur=spec.cursor()
343                 query = "select * from course;"
344                 mycur.execute(query)
345                 data = mycur.fetchall()

```

```

346         #data = self.db_connection.select(query)
347         # print(data)
348         # self.view_course_tree.delete(*self.view_course_tree.get_children())
349         for values in data:
350             data_list = [values[0], values[1], values[2], values[3]]
351             # print(data_list)
352             view_course_tree.insert('', END, values=data_list)
353
354     except BaseException as msg:
355         print(msg)
356     view_course_frame = Toplevel() #Frame(view_frame, bg="white")
357     view_course_frame.geometry("800x500")
358     #view_course_frame.place(x=10, y=320, height=250, width=575)
359
360     scroll_y_e = Scrollbar(view_course_frame, orient=VERTICAL)
361     scroll_x_e = Scrollbar(view_course_frame, orient=HORIZONTAL)
362     view_course_tree = ttk.Treeview(view_course_frame,columns=("COURSE ID", "COURSE
NAME", "COURSE DURATION", "COURSE CREDIT"),xscrollcommand=scroll_x_e.set, yscrollcommand=
scroll_y_e.set)
363         scroll_x_e.pack(side=BOTTOM, fill=X)
364         scroll_y_e.pack(side=RIGHT, fill=Y)
365         scroll_x_e.config(command=view_course_tree.xview)
366         scroll_y_e.config(command=view_course_tree.yview)
367
368     # =====TreeView Heading=====
369     view_course_tree.heading("COURSE ID", text="COURSE ID")
370     view_course_tree.heading("COURSE NAME", text="COURSE NAME")
371     view_course_tree.heading("COURSE DURATION", text="COURSE DURATION")
372     view_course_tree.heading("COURSE CREDIT", text="COURSE CREDIT")
373     view_course_tree["show"] = "headings"
374
375     # =====TreeView Column=====
376     view_course_tree.column("COURSE ID", width=50)
377     view_course_tree.column("COURSE NAME", width=150)
378     view_course_tree.column("COURSE DURATION", width=100)
379     view_course_tree.column("COURSE CREDIT", width=100)
380     view_course_tree.pack(fill=BOTH, expand=1)
381
382     view_course_information()
383
384     def dep_info():
385         # =====
386         # =====Displaying Department Information=====
387         # =====
388         def view_department_information():
389             """fetched data of department from database and inserted required index to
department tree view"""
390             try:
391                 #obj_student_database = Model_class.student_registration.GetDatabase('use
cms;')
392                 #self.db_connection.create(obj_student_database.get_database())
393                 a=load_data()
394                 host=a[0]
395                 username = a[2]
396                 password = a[3]
397                 port=a[1]
398
399                 spec=sql.connect(host=host,user=username,password=password,port=port,
database="sms")
400                 mycur=spec.cursor()
401                 query = "select * from department;"
402                 mycur.execute(query)
403                 data = mycur.fetchall()

```

```

404
405         for values in data:
406             data_list = [values[0], values[2], values[1]]
407             # print(data_list)
408             view_department_tree.insert('', END, values=data_list)
409
410     except BaseException as msg:
411         print(msg)
412
413     view_department_frame = Toplevel()
414     view_department_frame.geometry("800x500") #Frame(view_frame, bg="white")
415     #view_department_frame.place(x=595, y=320, height=250, width=575)
416
417     scroll_y_e = Scrollbar(view_department_frame, orient=VERTICAL)
418     scroll_x_e = Scrollbar(view_department_frame, orient=HORIZONTAL)
419     view_department_tree = ttk.Treeview(view_department_frame, columns=("DEPARTMENT ID"
420 , "DEPARTMENT NAME", "DEPARTMENT CODE"), xscrollcommand=scroll_x_e.set, yscrollcommand=
421 scroll_y_e.set)
422     scroll_x_e.pack(side=BOTTOM, fill=X)
423     scroll_y_e.pack(side=RIGHT, fill=Y)
424     scroll_x_e.config(command=view_department_tree.xview)
425     scroll_y_e.config(command=view_department_tree.yview)
426
427     # =====TreeView Heading=====
428     view_department_tree.heading("DEPARTMENT ID", text="DEPARTMENT ID")
429     view_department_tree.heading("DEPARTMENT NAME", text="DEPARTMENT NAME")
430     view_department_tree.heading("DEPARTMENT CODE", text="DEPARTMENT CODE")
431     view_department_tree["show"] = "headings"
432
433     # =====TreeView Column=====
434     view_department_tree.column("DEPARTMENT ID", width=50)
435     view_department_tree.column("DEPARTMENT NAME", width=150)
436     view_department_tree.column("DEPARTMENT CODE", width=100)
437     view_department_tree.pack(fill=BOTH, expand=1)
438
439
440
441     student_view_label = Button(view_frame, text="View Students Information ", borderwidth=
442 0, activebackground="white", relief=FLAT, bg="white", fg="#4f4e4d", font=("yu gothic ui", 13, "bold"),
443 command=studentinfo)
444     student_view_label.place(x=70, y=60)
445
446     employee_view_label = Button(view_frame, text="View Employees Information ",
447 borderwidth=0, activebackground="white", relief=FLAT, bg="white", fg="#4f4e4d", font=("yu gothic
448 ui", 13, "bold"), command=employee_info)
449     employee_view_label.place(x=70, y=140)
450
451     department_view_label = Button(view_frame, text="View Department Information ",
452 borderwidth=0, activebackground="white", relief=FLAT, bg="white", fg="#4f4e4d", font=("yu gothic
453 ui", 13, "bold"), command=dep_info)
454     department_view_label.place(x=70, y=220)
455
456     course_view_label = Button(view_frame, text="View Course Information ", borderwidth=0,
457 activebackground="white", relief=FLAT, bg="white", fg="#4f4e4d", font=("yu gothic ui", 13, "bold
458 "), command=course_info)
459     course_view_label.place(x=70, y=300)
460
461
462     #view_student_information()
463     #view_employee_information()
464     #view_course_information()

```

```

457     #view_department_information()'''  

458  

459  

460  

461  

462     def click_exit():  

463         """ Allows user to terminates the program when chosen yes"""  

464         root.deiconify()  

465         ask = messagebox.askyesnocancel("Confirm Exit", "Are you sure you want to Exit\nStudent Registration Form?")  

466         if ask is True:  

467             root.quit()  

468  

469     def weather1():  

470         city="chennai Weather"  

471         city = city.replace(" ", "+")  

472  

473         res = requests.get(f'https://www.google.com/search?q={city}&oq={city}&aqs=chrome.0.  
35i39l2j0l4j46j69i60.6128j1j7&sourceid=chrome&ie=UTF-8', headers=headers)  

474         soup = BeautifulSoup(res.text,'html.parser')  

475         info = soup.select('#wob_dc')[0].getText().strip()  

476         weather = soup.select('#wob_tm')[0].getText().strip()  

477         a=weather+"°C"+" Chennai"  

478         concated_text = f" {info} \n {a} "  

479         weather_l.configure(text=concated_text, font=("yu gothic ui", 13, "bold"), relief=FLAT  

, borderwidth=0, background="white", foreground="black")  

480         weather_l.after(180000, weather1)  

481  

482  

483  

484  

485     def time_running():  

486         """ displays the current date and time which is shown at top left corner of admin  

dashboard"""  

487         #headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.  
36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.3'}  

488  

489  

490         dt=datetime.now()  

491         time1 = dt.strftime('%H:%M:%S') #time1 = datetime.strftime('%H:%M:%S')#  

492         date1 = date.today() #date = datetime.strftime('%Y/%m/%d')  

493         concated_text = f" {time1} \n {date1} " #concated_text = f" {time} \n {date}"  

494         date_time.configure(text=concated_text, font=("yu gothic ui", 13, "bold"), relief=FLAT  

, borderwidth=0, background="white", foreground="black")  

495         date_time.after(100, time_running)  

496  

497  

498  

499     root = Toplevel()  

500     root.geometry("1067x600")  

501     root.title("Dashboard")  

502     root.resizable(False, False)  

503     #root.iconbitmap('images\\logo.ico')  

504  

505     bg = ImageTk.PhotoImage(file="files\Dashboard1.jpg")  

506     lbl_bg = Label(root,image=bg)  

507     lbl_bg.image = bg  

508     lbl_bg.place(x=0,y=0,relwidth=1,relheight=1)  

509  

510     clock_image = ImageTk.PhotoImage(file="Pics/time.png")  

511     date_time_image = Label(root, image=clock_image, bg="white")  

512     date_time_image.image = clock_image  

513     date_time_image.place(x=32, y=50)

```

```

514
515     date_time = Label(root)
516     date_time.place(x=65, y=40)
517     time_running()
518
519     weather_image_r=Image.open("Pics/weather-news.png").resize((25,25),Image.ANTIALIAS)
520     weather_image = ImageTk.PhotoImage(weather_image_r)
521     weather_l_image = Label(root, image=weather_image, bg="white")
522     weather_l_image.image = weather_image
523     weather_l_image.place(x=160, y=50)
524
525     weather_l=Label(root)
526     weather_l.place(x=200,y=40)
527     weather1()
528
529
530     home_r=Image.open('Pics\\home.png')
531     home_r=home_r.resize((50, 50), Image.ANTIALIAS)
532     home = ImageTk.PhotoImage(home_r)
533     home_button = Button(root, image=home,font=("yu gothic ui", 13, "bold"), relief=FLAT,
      activebackground="white", borderwidth=0, background="white", cursor="hand2",command=click_home
    )
534     home_button.image = home
535     home_button.place(x=40, y=113)
536
537
538     manage_r=Image.open('Pics\\manage.png')
539     manage_r=manage_r.resize((50, 50), Image.ANTIALIAS)
540     manage = ImageTk.PhotoImage(manage_r)
541     manage_button = Button(root, image=manage,font=("yu gothic ui", 13, "bold"), relief=FLAT,
      activebackground="white", borderwidth=0, background="white", cursor="hand2",command=
        click_manage)
542     manage_button.image = manage
543     manage_button.place(x=38, y=205)
544
545     view_r=Image.open('Pics\\view.png')
546     view_r=view_r.resize((50, 50), Image.ANTIALIAS)
547     view = ImageTk.PhotoImage(view_r)
548     view_button = Button(root, image=view,font=("yu gothic ui", 13, "bold"), relief=FLAT,
      activebackground="white", borderwidth=0, background="white", cursor="hand2",command=click_view
    )
549     view_button.image = view
550     view_button.place(x=36, y=299)
551
552     setting_r = Image.open('Pics\\setting.png')
553     setting_r=setting_r.resize((50, 50), Image.ANTIALIAS)
554     setting = ImageTk.PhotoImage(setting_r)
555     setting_button = Button(root, image=setting,font=("yu gothic ui", 13, "bold"), relief=FLAT
      , activebackground="white", borderwidth=0, background="white", cursor="hand2")
556     setting_button.image = setting
557     setting_button.place(x=38, y=395)
558
559     exit_1_r=Image.open('Pics\\exit_button.png')
560     exit_1_r=exit_1_r.resize((50, 50), Image.ANTIALIAS)
561     exit_1 = ImageTk.PhotoImage(exit_1_r)
562     exit_button = Button(root, image=exit_1,font=("yu gothic ui", 13, "bold"), relief=FLAT,
      activebackground="white", borderwidth=0, background="white", cursor="hand2",command=click_exit
    )
563     exit_button.image = exit_1
564     exit_button.place(x=37, y=490)
565
566     logout = ImageTk.PhotoImage(file='Pics\\logout.png')
567     logout_button = Button(root, image=logout,font=("yu gothic ui", 13, "bold"), relief=FLAT,

```

```
567 activebackground="white", borderwidth=0, background="white", cursor="hand2", command=
  click_logout)
568     logout_button.image = logout
569     logout_button.place(x=940, y=56)
570
571
572
573     #root.mainloop()
574
575
576 if __name__=="__main__":
577     dashboard()
578
```