```python
from distutils import command
from tkinter import *
from tkinter import ttk
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 user_inter
import csv
import course_screen


def regist_emp():
    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 reg_as():
        """ to validate and register employee as different roles"""
        global department_combo
        try:
            #obj_employee_database = Model_class.employee_registration.GetDatabase('use cms;')
            #db_connection.create(obj_employee_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 department"
            mycur.execute(query)

            department_tuple = mycur.fetchall()
            # print(department_tuple)
            department_list = []
            for i in department_tuple:
                department_name = i[2]
                department_list.append(department_name)
                # print(department_name)
                # print(department_list)

        except BaseException as msg:
            print(msg)
        # ==========================================================================
        # ============================Department=================================
```

```
63          # ========================================================================
64          department_label = Label(personal_frame, text="Department ", bg="white", fg="#4f4e4d",
65                                   font=("yu gothic ui", 13, "bold"))
66          department_label.place(x=370, y=170)  # (x=370, y=130) (x=370, y=170)
67
68          department_combo = ttk.Combobox(root, font=('yu gothic ui semibold', 12, 'bold'),
    state='readonly',
69                                          width=31)
70
71          department_combo['values'] = department_list
72          try:
73              department_combo.current(0)
74          except:
75              messagebox.showerror("Error", "You must add Department first")
76          department_combo.place(x=605, y=437)
77
78
79      def validation():
80              """this will validate if the username and email of entry fields are already in
    database table named employee or
81              not if return True, error message is thrown displaying email/username already
    exists"""
82              try:
83                  #obj_student_database = Model_class.employee_registration.GetDatabase('use cms
    ;')
84                  #db_connection.create(obj_student_database.get_database())
85                  a=load_data()
86                  host=a[0]
87                  username = a[2]
88                  password = a[3]
89                  port=a[1]
90
91                  spec=sql.connect(host=host,user=username,password=password,port=port,database=
    "sms")
92                  mycur=spec.cursor()
93
94                  query = "select * from employees;"
95                  mycur.execute(query)
96                  data = mycur.fetchall()
97                  # print(data)
98                  username_list = []
99                  email_list = []
100                 for values in data:
101                     # print(values)
102                     user_data_list = values[1]
103                     username_list.append(user_data_list)
104                     email_data_list = values[2]
105                     email_list.append(email_data_list)
106                     # print(final_list)
107                     # print(data_list)
108             except BaseException as msg:
109                 print(msg)
110
111             if username_entry.get() == "" or email_entry.get() == "" or password_entry.get
    () == "" \
112                     or f_name_entry.get() == "" or l_name_entry.get() == "" or dob_entry.get
    () == "" \
113                     or address_entry.get() == "" or contact_entry.get() == "":
114                 messagebox.showwarning("Warning", "All Fields are Required\n Please fill all
    required fields")
115
116             elif username_entry.get() in username_list:
117                 messagebox.showerror("Already Exists", f"{username_entry.get()} username
```

```
117 Already Exists")
118 
119             elif "@" not in email_entry.get():
120                 messagebox.showerror("Invalid Email", f"{email_entry.get()} Email ID Invalid")
121 
122             elif email_entry.get() in email_list:
123                 messagebox.showerror("Already Exists", f"{email_entry.get()} Email ID Already
    Exists")
124 
125             elif password_entry.get() != c_password_entry.get():
126                 messagebox.showerror("Not Matched", "Password Does not Matched")
127 
128             else:
129                 click_submit()
130     def back():
131         root.destroy()
132 
133     def click_submit():
134         """initialize when click submit button, which will take data from entry box
135         and insert those data into employee table after successful validation of those data"""
136         try:
137             #obj_employee_database = Model_class.employee_registration.GetDatabase('use cms;')
138             #db_connection.create(obj_employee_database.get_database())
139             a=load_data()
140             host=a[0]
141             username = a[2]
142             password = a[3]
143             port=a[1]
144 
145             spec=sql.connect(host=host,user=username,password=password,port=port,database="sms
    ")
146             mycur=spec.cursor()
147 
148             #obj_employee_database = Model_class.employee_registration.EmployeeRegistration(
    username_entry.get(),
149                                                                                            #
    email_entry.get(),
150                                                                                            #
    password_entry.get(),
151                                                                                            #
    f_name_entry.get(),
152                                                                                            #
    l_name_entry.get(),
153                                                                                            #
    dob_entry.get(),
154                                                                                            #
    gender_combo.get(),
155                                                                                            #
    address_entry.get(),
156                                                                                            #
    contact_entry.get(),
157                                                                                            #
    job_type_combo.get(),
158                                                                                            #
    register_as_combo.get(),
159                                                                                            #
    qualification_entry.get(),
160                                                                                            #
    department_combo.get(),
161                                                                                            #
    reg_date)
162             query = f"insert into employees (username,email,password,f_name,l_name,dob,gender,
    address," \
```

```
163                     f"contact_no,job_type,registered_as,qualification,department,reg_date)
    values ('{username_entry.get()}','{email_entry.get()}','{password_entry.get()}','{f_name_entry
    .get()}','{l_name_entry.get()}','{dob_entry.get()}'," \
164                     f"'{gender_combo.get()}','{address_entry.get()}','{contact_entry.get()}','
    {job_type_combo.get()}','{register_as_combo.get()}','{qualification_entry.get()}','{
    department_combo.get()}','{reg_date}');"
165
166
167             #values = (obj_employee_database.get_username(), obj_employee_database.get_email
    (),
168                         #obj_employee_database.get_password(), obj_employee_database.
    get_f_name(),
169                         #obj_employee_database.get_l_name(), obj_employee_database.get_dob(),
170                         #obj_employee_database.get_gender(), obj_employee_database.get_address
    (),
171                         #obj_employee_database.get_contact(), obj_employee_database.get_job(),
172                         #obj_employee_database.get_reg_as(), obj_employee_database.
    get_qualification(),
173                         #obj_employee_database.get_department(), obj_employee_database.
    get_reg_date())
174             # print(values)
175             #db_connection.insert(query, values)
176             # print(values)
177             mycur.execute(query)
178             spec.commit()
179             messagebox.showinfo("Success", f"Data inserted Successfully\n Employee name={
    f_name_entry.get()},\n "
180                                             f"JOB={job_type_combo.get()}")
181
182         except BaseException as msg:
183             print(msg)
184             messagebox.showerror("Error", "There is some error Submitting Credentials ")
185
186     def exit():
187         ask = messagebox.askyesnocancel("Confirm Exit", "Are you sure you want to Exit\n
    College Management System?")
188         if ask is True:
189             root.destroy()
190
191
192     root = Toplevel()
193     root.title('EMPLOYEE MANAGEMENT SYSTEM - COLLEGE MANAGEMENT SYSTEM')
194     root.geometry('1067x600')
195     root.config(bg="#f29844")
196
197     # ======================Backend connection=============
198     #db_connection = Backend.connection.DatabaseConnection()
199
200     reg_frame = Frame(root, bg="#ffffff", width=1000, height=560)
201     reg_frame.place(x=30, y=30)
202
203
204     heading = Label(reg_frame, text="Employee Registration Form", font=('yu gothic ui', 20, "
    bold"), bg="white",
205                             fg='black',
206                             bd=5,
207                             relief=FLAT)
208     heading.place(x=200, y=0, width=600)
209     #slider()
210     #heading_color()
211
212     cred_frame = LabelFrame(reg_frame, text="Account Details", bg="white", fg="#4f4e4d",
    height=140,
```

```
                                    width=800, borderwidth=2.4,
                                    font=("yu gothic ui", 13, "bold"))
    cred_frame.config(highlightbackground="red")
    cred_frame.place(x=100, y=50)

    # ============================================================================
    # ============================Username========================================
    # ============================================================================

    username_label = Label(cred_frame, text="Username ", bg="white", fg="#4f4e4d",
                                font=("yu gothic ui", 13, "bold"))
    username_label.place(x=10, y=10)

    username_entry = Entry(root, highlightthickness=0, relief=FLAT, bg="white", fg="#6b6a69",
                                font=("yu gothic ui semibold", 12))
    username_entry.place(x=230, y=117, width=260)  # trebuchet ms

    username_line = Canvas(root, width=260, height=1.5, bg="#bdb9b1", highlightthickness=0)
    username_line.place(x=230, y=139)

    # ============================================================================
    # ============================Email===========================================
    # ============================================================================

    email_label = Label(cred_frame, text="Email ", bg="white", fg="#4f4e4d",
                                font=("yu gothic ui", 13, "bold"))
    email_label.place(x=370, y=10)

    email_entry = Entry(root, highlightthickness=0, relief=FLAT, bg="white", fg="#6b6a69",
                                font=("yu gothic ui semibold", 12))
    email_entry.place(x=555, y=117, width=350)  # trebuchet ms

    email_line = Canvas(root, width=350, height=1.5, bg="#bdb9b1", highlightthickness=0)
    email_line.place(x=555, y=139)

    # ============================================================================
    # ============================Password========================================
    # ============================================================================

    password_label = Label(cred_frame, text="Password ", bg="white", fg="#4f4e4d",
                                font=("yu gothic ui", 13, "bold"))
    password_label.place(x=10, y=50)

    password_entry = Entry(root, highlightthickness=0, relief=FLAT, bg="white", fg="#6b6a69",
                                font=("yu gothic ui semibold", 12),show = "*")
    password_entry.place(x=230, y=157, width=260)  # trebuchet ms

    password_line = Canvas(root, width=260, height=1.5, bg="#bdb9b1", highlightthickness=0)
    password_line.place(x=230, y=180)

    # ============================================================================
    # ============================Confirm password================================
    # ============================================================================

    c_password_label = Label(cred_frame, text="Confirm Password ", bg="white", fg="#4f4e4d",
                                    font=("yu gothic ui", 13, "bold"))
    c_password_label.place(x=370, y=50)

    c_password_entry = Entry(root, highlightthickness=0, relief=FLAT, bg="white", fg="#6b6a69"
,
                                    font=("yu gothic ui semibold", 12), show = "*")
    c_password_entry.place(x=650, y=157, width=255)  # trebuchet ms

```

```
275     c_password_line = Canvas(root, width=255, height=1.5, bg="#bdb9b1", highlightthickness=0)
276     c_password_line.place(x=650, y=180)
277 
278     # ============================================================================
279     # ========================frame for personal credentails=====================
280     # ============================================================================
281 
282     personal_frame = LabelFrame(reg_frame, text="Personal Details", bg="white", fg="#4f4e4d",
    height=265,
283                                         width=800, borderwidth=2.4,
284                                         font=("yu gothic ui", 13, "bold"))
285     personal_frame.config(highlightbackground="red")
286     personal_frame.place(x=100, y=210)
287 
288 
289     # ============================================================================
290     # ============================First name======================================
291     # ============================================================================
292     f_name_label = Label(personal_frame, text="First Name ", bg="white", fg="#4f4e4d",
293                                 font=("yu gothic ui", 13, "bold"))
294     f_name_label.place(x=10, y=10)
295 
296     f_name_entry = Entry(root, highlightthickness=0, relief=FLAT, bg="white", fg="#6b6a69",
297                                 font=("yu gothic ui semibold", 12))
298     f_name_entry.place(x=235, y=277, width=260)  # trebuchet ms
299 
300     f_name_line = Canvas(root, width=260, height=1.5, bg="#bdb9b1", highlightthickness=0)
301     f_name_line.place(x=235, y=299)
302 
303     # ============================================================================
304     # ============================Last name=======================================
305     # ============================================================================
306 
307     l_name_label = Label(personal_frame, text="Last Name ", bg="white", fg="#4f4e4d",
308                                 font=("yu gothic ui", 13, "bold"))
309     l_name_label.place(x=370, y=10)
310 
311     l_name_entry = Entry(root, highlightthickness=0, relief=FLAT, bg="white", fg="#6b6a69",
312                                 font=("yu gothic ui semibold", 12))
313     l_name_entry.place(x=595, y=277, width=315)  # trebuchet ms
314 
315     l_name_line = Canvas(root, width=315, height=1.5, bg="#bdb9b1", highlightthickness=0)
316     l_name_line.place(x=595, y=299)
317 
318     # ============================================================================
319     # ============================DOB=============================================
320     # ============================================================================
321 
322     dob_label = Label(personal_frame, text="DOB ", bg="white", fg="#4f4e4d",
323                               font=("yu gothic ui", 13, "bold"))
324     dob_label.place(x=10, y=50)
325 
326     dob_entry = Entry(root, highlightthickness=0, relief=FLAT, bg="white", fg="#6b6a69",
327                               font=("yu gothic ui semibold", 12))
328     dob_entry.insert(0, "mm/dd/yyyy")
329     dob_entry.place(x=190, y=317, width=305)  # trebuchet ms
330     #dob_entry.bind("<1>", pick_date)
331 
332     dob_line = Canvas(root, width=305, height=1.5, bg="#bdb9b1", highlightthickness=0)
333     dob_line.place(x=190, y=339)
334 
335     # ============================================================================
336     # ============================Gender==========================================
```

```
337     # ========================================================================
338     style = ttk.Style()
339 
340     # style.map('TCombobox', selectbackground=[('readonly', 'grey')])
341     root.option_add("*TCombobox*Listbox*Foreground", '#f29844')
342 
343     gender_label = Label(personal_frame, text="Gender ", bg="white", fg="#4f4e4d",
344                                 font=("yu gothic ui", 13, "bold"))
345     gender_label.place(x=370, y=50)
346 
347     gender_combo = ttk.Combobox(root, font=('yu gothic ui semibold', 12, 'bold'), state='
    readonly',
348                                         width=35)
349     gender_combo['values'] = ('Male', 'Female', 'Rather not say')
350     gender_combo.current(0)
351     gender_combo.place(x=570, y=317)
352 
353     # gender_line = Canvas(root, width=315, height=1.5, bg="#bdb9b1", highlightthickness=0)
354     # gender_line.place(x=595, y=369)
355     #aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
356     # ========================================================================
357     # ============================Address=====================================
358     # ========================================================================
359 
360     address_label = Label(personal_frame, text="Address ", bg="white", fg="#4f4e4d",
361                                 font=("yu gothic ui", 13, "bold"))
362     address_label.place(x=10, y=90)
363 
364     address_entry = Entry(root, highlightthickness=0, relief=FLAT, bg="white", fg="#6b6a69",
365                                 font=("yu gothic ui semibold", 12))
366     address_entry.place(x=215, y=357, width=280)  # trebuchet ms
367 
368     address_line = Canvas(root, width=280, height=1.5, bg="#bdb9b1", highlightthickness=0)
369     address_line.place(x=215, y=379)
370 
371     # ========================================================================
372     # ============================Contact no==================================
373     # ========================================================================
374 
375     contact_label = Label(personal_frame, text="Contact No. ", bg="white", fg="#4f4e4d",
376                                 font=("yu gothic ui", 13, "bold"))
377     contact_label.place(x=370, y=90)
378 
379     contact_entry = Entry(root, highlightthickness=0, relief=FLAT, bg="white", fg="#6b6a69",
380                                 font=("yu gothic ui semibold", 12))
381     contact_entry.place(x=605, y=357, width=305)  # trebuchet ms
382 
383     contact_line = Canvas(root, width=305, height=1.5, bg="#bdb9b1", highlightthickness=0)
384     contact_line.place(x=605, y=379)
385 
386     # ========================================================================
387     # ============================Job Type====================================
388     # ========================================================================
389 
390     job_type_label = Label(personal_frame, text="Job Type ", bg="white", fg="#4f4e4d",
391                                 font=("yu gothic ui", 13, "bold"))
392     job_type_label.place(x=10, y=130)
393 
394     job_type_combo = ttk.Combobox(root, font=('yu gothic ui semibold', 12, 'bold'), state='
    readonly',
395                                         width=27)
396     job_type_list = ["Part time", "Full time"]
397     job_type_combo['values'] = job_type_list
```

```
398     job_type_combo.current(0)
399     job_type_combo.place(x=223, y=397)
400 
401     # ============================================================================
402     # ============================Register as=====================================
403     # ============================================================================
404 
405     register_as_label = Label(personal_frame, text="Register as ", bg="white", fg="#4f4e4d",
406                                       font=("yu gothic ui", 13, "bold"))
407     register_as_label.place(x=370, y=130)
408 
409     register_as_combo = ttk.Combobox(root, font=('yu gothic ui semibold', 12, 'bold'), state='
    readonly',
410                                            width=31)
411 
412     register_as_list = ["Department Head", "Instructor", "Employee"]
413     register_as_combo['values'] = register_as_list
414     register_as_combo.current(2)
415     register_as_combo.place(x=605, y=397)
416     #register_as_combo.bind('<<ComboboxSelected>>', reg_as_event_handle)
417 
418     # ============================================================================
419     # ============================Qualification=================================
420     # ============================================================================
421 
422     qualification_label = Label(personal_frame, text="Qualification ", bg="white", fg="#4f4e4d
    ",
423                                         font=("yu gothic ui", 13, "bold"))
424     qualification_label.place(x=10, y=170)
425 
426     qualification_entry = Entry(root, highlightthickness=0, relief=FLAT, bg="white", fg="#
    6b6a69",
427                                         font=("yu gothic ui semibold", 12))
428     qualification_entry.place(x=250, y=437, width=240)  # trebuchet ms
429 
430     qualification_line = Canvas(root, width=240, height=1.5, bg="#bdb9b1", highlightthickness=
    0)
431     qualification_line.place(x=250, y=460)
432 
433     reg_date = time.strftime("%Y/%m/%d")
434 
435     reg_as()
436 
437     # ============================================================================
438     # ============================Register options================================
439     # ============================================================================
440 
441     options_frame = LabelFrame(reg_frame, text="Register Options", bg="white", fg="#4f4e4d",
    height=80,
442                                       width=800, borderwidth=2.4,
443                                       font=("yu gothic ui", 13, "bold"))
444     options_frame.config(highlightbackground="red")
445     options_frame.place(x=100, y=475)
446 
447     # ============================================================================
448     # ============================Register options================================
449     # ============================================================================
450 
451     submit_img = ImageTk.PhotoImage(file='Pics\\submit.png')
452     submit = Button(options_frame, image=submit_img,
453                               font=("yu gothic ui", 13, "bold"), relief=FLAT, activebackground="
    white"
454                               , borderwidth=0, background="white", cursor="hand2",command=
```

```
validation)
    submit.image = submit_img
    submit.place(x=90, y=10)

    clear_img = ImageTk.PhotoImage(file='Pics\\clear.png')
    clear_button = Button(options_frame, image=clear_img,
                                font=("yu gothic ui", 13, "bold"), relief=FLAT,
activebackground="white"
                                , borderwidth=0, background="white", cursor="hand2")
    clear_button.image = clear_img
    clear_button.place(x=250, y=13)

    back_img = ImageTk.PhotoImage(file='Pics\\back.png')
    back_button = Button(options_frame, image=back_img,
                                font=("yu gothic ui", 13, "bold"), relief=FLAT,
activebackground="white"
                                , borderwidth=0, background="white", cursor="hand2",command=
back)
    back_button.image = back_img
    back_button.place(x=410, y=13)

    exit_img = ImageTk.PhotoImage(file='Pics\\exit.png')
    exit_button = Button(options_frame, image=exit_img,
                                font=("yu gothic ui", 13, "bold"), relief=FLAT,
activebackground="white"
                                , borderwidth=0, background="white", cursor="hand2", command=
exit)
    exit_button.image = exit_img
    exit_button.place(x=570, y=13)

    #root.mainloop()


if __name__ == "__main__":
    regist_emp()
```