

NAME: SHRADDHA HIRAMAN BONDEKAR

BATCH: PYT42

DATE: 4/1/2022

CODE:

```
from tkinter import*
from tkinter import ttk
from PIL import Image, ImageTk
import mysql.connector
from tkinter import messagebox
class Employee:
    def __init__(self, root):
        self.root=root
        self.root.geometry("1530x790+0+0")
        self.root.title('Employee Management System')

        #variable
        self.var_dep=StringVar()
        self.var_name=StringVar()
        self.var_designition=StringVar()
        self.var_email=StringVar()
        self.var_address=StringVar()
        self.var_married=StringVar()
        self.var_dob=StringVar()
        self.var_doj=StringVar()
        self.var_idproofcomb=StringVar()
        self.var_idproof=StringVar()
        self.var_gender=StringVar()
        self.var_phone=StringVar()
        self.var_country=StringVar()
        self.var_salary=StringVar()

        lbl_title=Label(self.root, text="EMPLOYEE MANAGEMENT
SYSTEM", font=('times new roman', 37, 'bold'), fg='darkblue', bg='white')
        lbl_title.place(x=0, y=0, width=1530, height=50)

        #logo
        img_logo=Image.open("C:/Users/shraddha/OneDrive/Desktop/logo.jpg")
        img_logo=img_logo.resize((50, 50), Image.ANTIALIAS)
        self.photo_logo=ImageTk.PhotoImage(img_logo)

        self.logo=Label(self.root, image=self.photo_logo)
        self.logo.place(x=270, y=0, width=50, height=50)
        #Image Frame
        img_frame=Frame(self.root, bd=2, relief=RIDGE, bg="white")
        img_frame.place(x=0, y=50, width=1530, height=160)

        #1st image
        img1=Image.open("C:/Users/shraddha/OneDrive/Desktop/img1.jpg")
        img1=img1.resize((540, 160), Image.ANTIALIAS)
        self.photo1=ImageTk.PhotoImage(img1)
```

```

self.img_1=Label(img_frame,image=self.photo1)
self.img_1.place(x=0, y=0)

#2nd image
img_2=Image.open(r"C:\Users\shraddha\OneDrive\Desktop\image2.jpg")
img_2=img_2.resize((540,160), Image.ANTIALIAS)
self.photo2=ImageTk.PhotoImage(img_2)

self.img_2= Label(img_frame,image=self.photo2)
self.img_2.place(x=540, y=0)

#3rd image
img_3=Image.open(r"C:\Users\shraddha\OneDrive\Desktop\image3.jpg")
img_3=img_3.resize((540,160), Image.ANTIALIAS)
self.photo3=ImageTk.PhotoImage(img_3)

self.img_3=Label(self.root, image=self.photo3)
self.img_3.place(x=1050, y=50)

# Main Frame
Main_frame=Frame(self.root, bd=2, relief=RIDGE, bg="white")
Main_frame.place(x=10,y=210,width=1500,height=560)

# Upper Frame
upper_frame=LabelFrame(Main_frame,bd=2, relief=RIDGE,
bg="white",text="Employee Information",font=('times new
roman',11,'bold'),fg='red')
upper_frame.place(x=10,y=10,width=1480,height=270)

# Labels and entry

lbl_dep=Label(upper_frame,text='Department',font=('arial',11,'bold'),fg='black')
lbl_dep.grid(row=0,column=0,padx=2,sticky=W)

combo_dept=ttk.Combobox(upper_frame,textvariable=self.var_dep,font=('arial',12,'bold'),width=17,state='readonly')
combo_dept['value']=('Select Department','HR','Software Engineer','Manager')
combo_dept.current(0)
combo_dept.grid(row=0,column=1,padx=2,pady=10,sticky=W)
#Name
lbl_name= Label(upper_frame,text='Name',font=('arial', 12,'bold'),fg='black')
lbl_name.grid(row=0, column=2, padx=2,pady=7,sticky=W)

txt_name=ttk.Entry(upper_frame,textvariable=self.var_name,width=22,font=('arial',11,'bold'))
txt_name.grid(row=0,column=3,padx=2,pady=7,sticky=W)

#lbl_Designition
lbl_Designition= Label(upper_frame,font=('arial', 12,'bold'),text='Designition:', fg='black')
lbl_Designition.grid(row=1,column=0, padx=2, pady=7, sticky=W)

txt_Designition=
ttk.Entry(upper_frame,textvariable=self.var_designition, width=22,font=('arial', 11, 'bold'))

```

```

txt_Designition.grid(row=1, column=1, padx=2, pady=7,sticky=W)

#Email
lbl_email= Label(upper_frame, font=('arial', 12, 'bold'),
text='Email:', fg='black')
lbl_email.grid(row=1, column=2, padx=2, pady=7, sticky=W)

txt_email= ttk.Entry(upper_frame,textvariable=self.var_email,
width=22, font=('arial', 11, 'bold'))
txt_email.grid(row=1, column=3,padx=2, pady=7)

#address
lbl_address= Label(upper_frame, font=('arial', 11, 'bold'),
text='Address:', fg='black')
lbl_address.grid(row=2, column=0, padx=2, pady=7, sticky=W)

txt_name = ttk.Entry(upper_frame,textvariable=self.var_address,
width=22, font=('arial', 11, 'bold'))
txt_name.grid(row=2, column=1, padx=2, pady=7,sticky=W)

#married
lbl_married_status= Label(upper_frame, font=('arial', 12, 'bold'),
text='Married Status', fg='black')
lbl_married_status.grid(row=2, column=2, padx=2, pady=7, sticky=W)

combo_txt_married=
ttk.Combobox(upper_frame,textvariable=self.var_married, font=('arial', 12,
'bold'), width=17, state='readonly')
combo_txt_married['value'] = ("Married","Unmarried","Single")
combo_txt_married.current(0)
combo_txt_married.grid(row=2, column=3, padx=2, pady=7, sticky=W)

#DOB
lbl_dob= Label(upper_frame, font=('arial', 12, 'bold'),
text='DOB:', fg='black')
lbl_dob.grid(row=3, column=0, padx=2, pady=7, sticky=W)

txt_dob= ttk.Entry(upper_frame,textvariable=self.var_dob, width=22,
font=('arial', 11, 'bold'))
txt_dob.grid(row=3, column=1, padx=2, pady=7)

# DOJ
lbl_doj = Label(upper_frame, font=('arial', 12, 'bold'),
text='DOJ:', fg='black')
lbl_doj.grid(row=3, column=2, padx=2, pady=7, sticky=W)

txt_doj = ttk.Entry(upper_frame,textvariable=self.var_doj,
width=22, font=('arial', 11, 'bold'))
txt_doj.grid(row=3, column=3, padx=2, pady=7)

#ID proof

com_txt_proof=ttk.Combobox(upper_frame,textvariable=self.var_idproof,state=
"readonly",font=('arial', 12, 'bold'),width=17)
com_txt_proof['value']=("Select ID Proof","PAN CARD","ADHAR
CARD","DRIVING LICENCE")
com_txt_proof.current(0)
com_txt_proof.grid(row=4,column=1,sticky=W,padx=2,pady=7)

lbl_proof= Label(upper_frame, font=('arial', 12, 'bold'),

```

```

text='ID_Proof:', fg='black')
lbl_proof.grid(row=4, column=0, padx=2, pady=7, sticky=W)

#gender
lbl_gender= Label(upper_frame, font=('arial', 12, 'bold'),
text="Gender:", fg='black')
lbl_gender.grid(row=4, column=2, padx=2, pady=7, sticky=W)

com_txt_gender =
ttk.Combobox(upper_frame,textvariable=self.var_gender, state="readonly",
font=('arial', 12, 'bold'), width=17)
com_txt_gender['value'] = ("Male","Female","Other")
com_txt_gender.current(0)
com_txt_gender.grid(row=4, column=3, sticky=W, padx=2, pady=7)

#phone no
lbl_phone= Label(upper_frame, font=('arial', 12, 'bold'),
text='Phone No:', fg='black')
lbl_phone.grid(row=0, column=4, padx=2, pady=7, sticky=W)

txt_phone= ttk.Entry(upper_frame,textvariable=self.var_phone,
width=22, font=('arial', 11, 'bold'))
txt_phone.grid(row=0, column=5, padx=2, pady=7)

#country
lbl_country= Label(upper_frame, font=('arial', 12, 'bold'),
text='Country:', fg='black')
lbl_country.grid(row=1, column=4, padx=2, pady=7, sticky=W)

txt_country = ttk.Entry(upper_frame,textvariable=self.var_country,
width=22, font=('arial', 11, 'bold'))
txt_country.grid(row=1, column=5, padx=2, pady=7)

#CTC
lbl_ctc= Label(upper_frame, font=('arial', 12, 'bold'),
text='Salary(CTC):', fg='black')
lbl_ctc.grid(row=2, column=4, padx=2, pady=7, sticky=W)

txt_ctc= ttk.Entry(upper_frame,textvariable=self.var_salary,
width=22, font=('arial', 11, 'bold'))
txt_ctc.grid(row=2, column=5, padx=2, pady=7)

#mask image
img_mask=
Image.open("C:/Users/shraddha/OneDrive/Desktop/maskimage.jpg")
img_mask=img_mask.resize((220,220), Image.ANTIALIAS)
self.photomask= ImageTk.PhotoImage(img_mask)

self.img_mask= Label(upper_frame, image=self.photomask)
self.img_mask.place(x=1000, y=0, width=220,height=220)

# Button Frame
button_frame=Frame(upper_frame, bd=2, relief=RIDGE, bg="white")
button_frame.place(x=1290, y=10, width=170, height=220)

btn_add=Button(button_frame,text="Save",command=self.add_data,font=("arial"
,15,"bold"),width=15,bg="blue",fg="white")
btn_add.grid(row=0,column=0,padx=1,pady=5)

btn_update= Button(button_frame,

```

```

text="Update",command=self.update_data,font=("arial", 15, "bold"),
width=15, bg="blue", fg="white")
    btn_update.grid(row=1, column=0, padx=1, pady=5)

    btn_delete = Button(button_frame,
text="Delete",command=self.delete_data, font=("arial", 15, "bold"),
width=15, bg="blue", fg="white")
    btn_delete.grid(row=2, column=0, padx=1, pady=5)

    btn_clear= Button(button_frame,
text="Clear",command=self.reset_data,font=("arial", 15, "bold"), width=15,
bg="blue", fg="white")
    btn_clear.grid(row=3, column=0, padx=1, pady=5)

    # Down Frame
    down_frame = LabelFrame(Main frame, bd=2, relief=RIDGE, bg="white",
text="Employee Table Information",font=('times new roman', 11,
'bold'),fg='blue')
    down_frame.place(x=10,y=280,width=1480,height=270)

    # Search Frame
    search_frame=LabelFrame(down_frame, bd=2, relief=RIDGE,text="Search
Employee Information",font=('times new roman', 11, 'bold'), fg='blue',
bg='white')
    search_frame.place(x=0, y=0, width=1470, height=60)

    search_by=Label(search_frame,font=("arial",11,"bold"),text="Search
By:",fg="white",bg="red")
    search_by.grid(row=0,column=0,sticky=W,padx=5)

    #search
    self.var_com_search=StringVar()

com_txt_search=ttk.Combobox(search_frame,textvariable=self.var_com_search,s
tate="readonly",font=("arial",12,"bold"),width=18)
    com_txt_search['value']=("Select Option","Phone","id_proof")
    com_txt_search.grid(row=0,column=1,sticky=W,padx=5)

    self.var_search=StringVar()

txt_search=ttk.Entry(search_frame,textvariable=self.var_search,width=22,font=
("arial",11,"bold"))
    txt_search.grid(row=0,column=2,padx=5)

btn_search=Button(search_frame,text="Search",command=self.search_data,font=
("arial",11,"bold"),width=14,bg="blue")
    btn_search.grid(row=0,column=3,padx=5)

    btn_ShowAll= Button(search_frame, text="Show All", font=("arial",
11, "bold"), width=14,bg="blue")
    btn_ShowAll.grid(row=0, column=4, padx=5)

    stayhome=Label(search_frame, text="Wear a Mask", font=("arial", 11,
"bold"), width=14,bg="red")
    stayhome.place(x=780,y=0,width=600,height=30)

# _____Employee table_____
    # Table Frame

    table_frame=Frame(down_frame, bd=3, relief=RIDGE, bg="white")

```

```

table_frame.place(x=0, y=60, width=1470, height=170)

scroll_x=ttk.Scrollbar(table_frame,orient=HORIZONTAL)
scroll_y=ttk.Scrollbar(table_frame,orient=VERTICAL)

self.employee_table=ttk.Treeview(table_frame,column=("dep","name","degi","e
mail","address","married","dob","doj","idproofcomb","idproof","gender","pho
ne","country","salary",),xscrollcommand=scroll_x.set,yscrollcommand=scroll_
y.set)

scroll_x.pack(side=BOTTOM,fill=X)
scroll_y.pack(side=RIGHT, fill=Y)

scroll_x.config(command=self.employee_table.xview)
scroll_y.config(command=self.employee_table.yview)

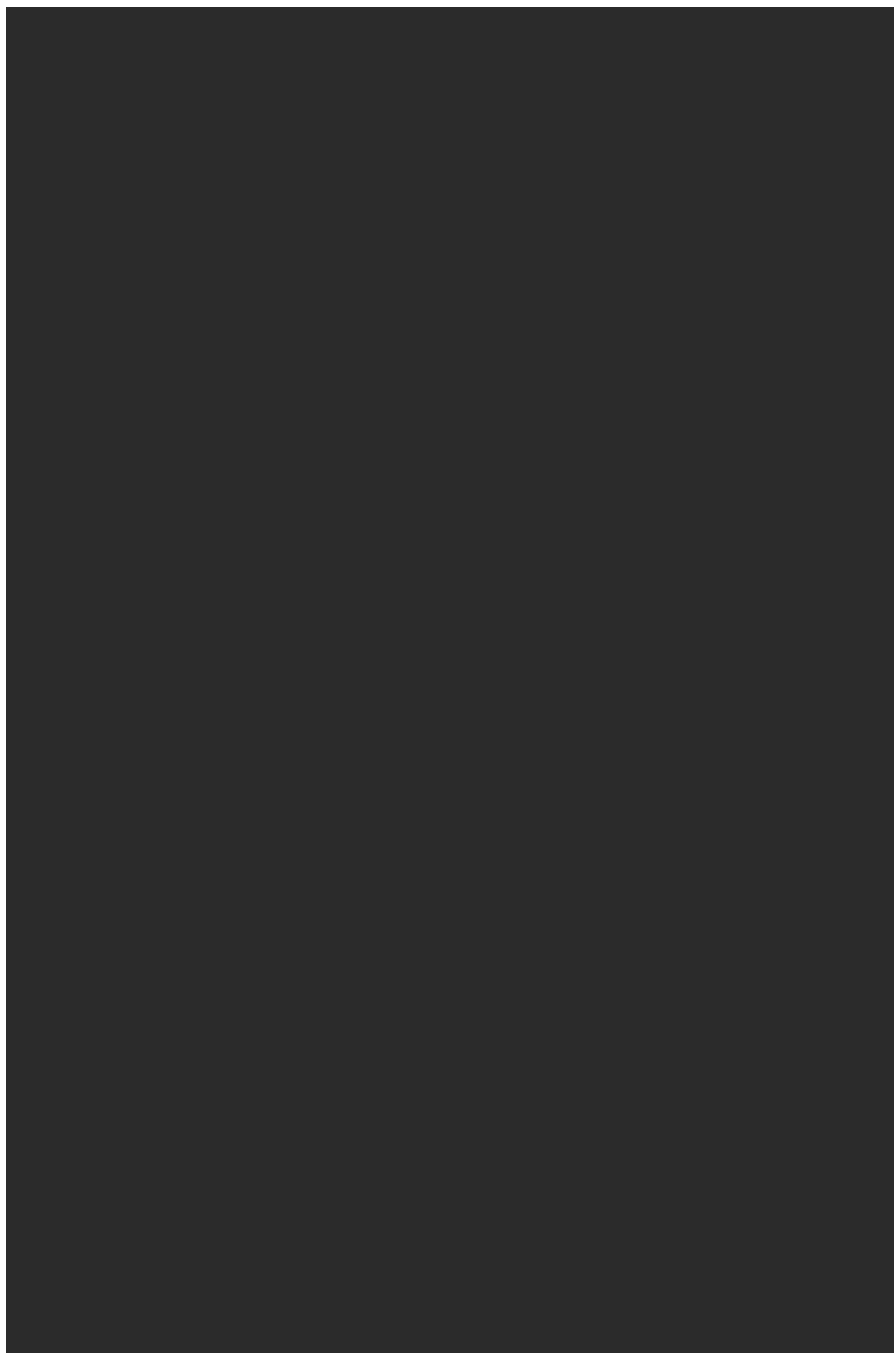
self.employee_table.heading('dep',text='Department')
self.employee_table.heading('name', text='Name')
self.employee_table.heading('degi', text='Degignition')
self.employee_table.heading('email', text='Email')
self.employee_table.heading('address', text='Address')
self.employee_table.heading('married', text='Married Status')
self.employee_table.heading('dob', text='DOB')
self.employee_table.heading('doj', text='DOJ')
self.employee_table.heading('idproofcomb', text='ID Type')
self.employee_table.heading('idproof', text='ID Proof')
self.employee_table.heading('gender', text='Gender')
self.employee_table.heading('phone', text='Phone')
self.employee_table.heading('country', text='Country')
self.employee_table.heading('salary', text='Salary')

self.employee_table['show']='headings'

self.employee_table.column("dep",width=100)
self.employee_table.column("name",width=100)
self.employee_table.column("degi",width=100)
self.employee_table.column("email",width=100)
self.employee_table.column("address",width=100)
self.employee_table.column("married",width=100)
self.employee_table.column("dob",width=100)
self.employee_table.column("doj",width=100)
self.employee_table.column("idproofcomb",width=100)
self.employee_table.column("idproof",width=100)
self.employee_table.column("gender",width=100)
self.employee_table.column("phone",width=100)
self.employee_table.column("country",width=100)
self.employee_table.column("salary",width=100)
self.fetch_data()
self.employee_table.pack(fill=BOTH,expand=1)
self.employee_table.bind("<ButtonRelease-1>",self.get_cursor)

#
#_____function Declaration_____
def add_data(self):
    if self.var_dep.get()==" " or self.var_email.get()==" ":
        messagebox.showerror("Error","All fields are requierd")
    else:
        try:

```



```
conn=mysql.connector.connect(host="localhost",user="root",password="shraddh  
abondekar@101020",database="mydata2")  
my_cursor=conn.cursor()
```



```

        my_cursor.execute('insert into employee3
values(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)', (

self.var_dep.get(),
self.var_name.get(),
self.var_designition.get(),
self.var_email.get(),
self.var_address.get(),
self.var_married.get(),
self.var_dob.get(),
self.var_doj.get(),
self.var_idproofcomb.get(),
self.var_idproof.get(),
self.var_gender.get(),
self.var_phone.get(),
self.var_country.get(),
self.var_salary.get()
))

        conn.commit()
        self.fetch_data()
        conn.close()
        messagebox.showinfo('Sucess','Employee has been
added...!!!',parent=self.root)
    except Exception as es:
        messagebox.showerror('Error',f'Due
To:{str(es)}',parent=self.root)

#fetch data
def fetch_data(self):
    conn = mysql.connector.connect(host="localhost", user="root",
password="shraddhabondekar@101020",database="mydata2")
    my_cursor = conn.cursor()
    my_cursor.execute("select * from employee3")
    data=my_cursor.fetchall()
    if len(data)!=0:
        # self.employee_table(*self.employee_table.get_children())
        for i in data:
            self.employee_table.insert("",END,values=i)
        conn.commit()
    conn.close()
# Get cursor
def get_cursor(self):
    cursor_row=self.employee_table.focus()
    content=self.employee_table.item(cursor_row)
    data=content['values']

    self.var_dep.set(data[0])

```

```

        self.var_name.set(data[1])
        self.var_designition.set(data[2])
        self.var_email.set(data[3])
        self.var_address.set(data[4])
        self.var_married.set(data[5])
        self.var_dob.set(data[6])
        self.var_doj.set(data[7])
        self.var_idproofcomb.set(data[8])
        self.var_idproof.set(data[9])
        self.var_gender.set(data[10])
        self.var_phone.set(data[11])
        self.var_country.set(data[12])
        self.var_salary.set(data[13])
#update
def update_data(self):
    if self.var_dep.get()==" " or self.var_email.get()==" ":
        messagebox.showerror("Error","All fields are requierd")
    else:
        try:
            update=messagebox.askyesno('Update','Are sure update this
employee data')
            if update>0:

conn=mysql.connector.connect(host="localhost",user="root",password="shraddh
abondekar@101020",database="mydata2")
            my_cursor=conn.cursor()
            my_cursor.execute('update employee3 set
dep=%s,name=%s,degi=%s,email=%s,address=%s,married=%s,dob=%s,idproofcomb=%s
,gender=%s,phone=%s,country=%s,salary=%s where idproof=%s',(
self.var_dep.get(),
self.var_name.get(),
self.var_designition.get(),
self.var_email.get(),
self.var_address.get(),
self.var_married.get(),
self.var_dob.get(),
self.var_doj.get(),
self.var_idproofcomb.get(),

self.var_gender.get(),
self.var_phone.get(),
self.var_country.get(),
self.var_salary.get(),
self.var_idproof.get()
))

```

```

        else:
            if not update:
                return
            conn.commit()
            self.fetch_data()
            conn.close()
            messagebox.showinfo('Success', 'Employee Successfully
updated', parent=self.root)
        except Exception as es:
            messagebox.showerror('Error', f'Due To:{str(es)}',
parent=self.root)

#delete
def delete_data(self):
    if self.var_dep.get() == "" or self.var_email.get() == "":
        messagebox.showerror("Error", "All fields are requierd")
    else:
        try:
            Delete=messagebox.askyesno('Delete', 'Are you sure delete
this employee', parent=self.root)
            if Delete>0:
                conn = mysql.connector.connect(host="localhost",
username="root", password="shraddhabondekar@101020",
database="mydata2")
                my_cursor = conn.cursor()
                sql='delete from employee3 where idproof=%s'
                value=(self.var_idproof.get(),)
                my_cursor.execute(sql,value)
            else:
                if not Delete:
                    return
                conn.commit()
                self.fetch_data()
                conn.close()
                messagebox.showinfo('Delete', 'Employee Successfully
Deleted', parent=self.root)
        except Exception as es:
            messagebox.showerror('Error', f'Due To:{str(es)}',
parent=self.root)

#reset
def reset_data(self):
    self.var_dep.set("Select Department")
    self.var_name.set("")
    self.var_designition.set("")
    self.var_email.set("")
    self.var_address.set("")
    self.var_married.set("Married")
    self.var_dob.set("")
    self.var_doj.set("")
    self.var_idproofcomb.set("Select ID Proof")
    self.var_idproof.set("")
    self.var_gender.set("")
    self.var_phone.set("")
    self.var_country.set("")
    self.var_salary.set("")

#search
def search_data(self):

```

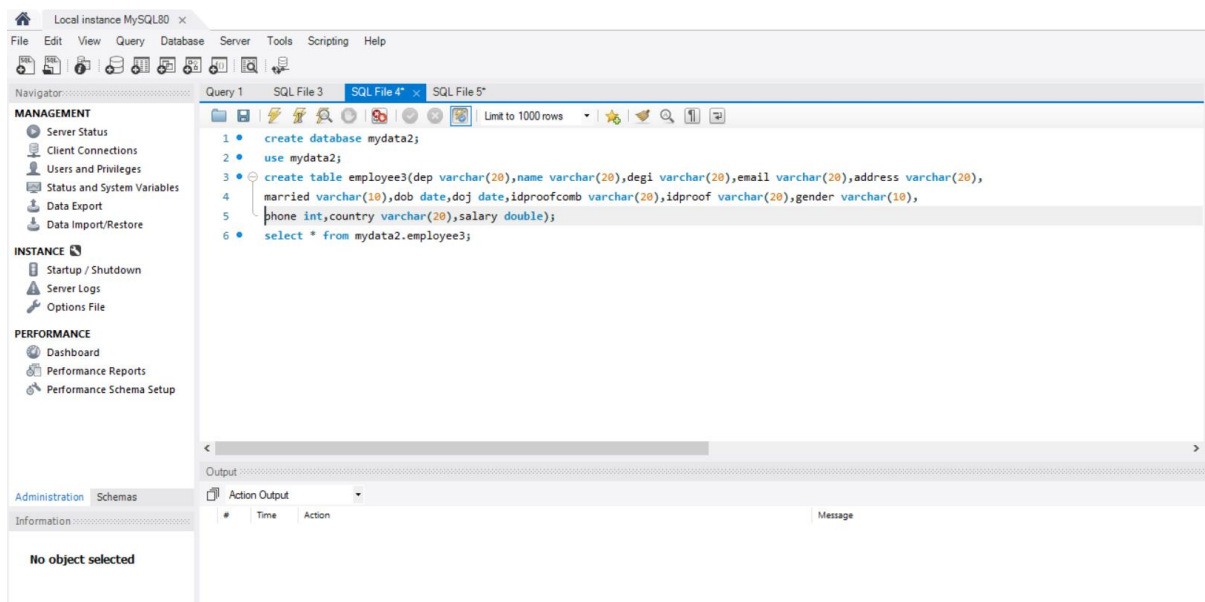
```

        if self.var_com_search.get()==' ' or self.var_search.get()==' ':
            messagebox.showerror('Error','Please select option')
        else:
            try:
                conn = mysql.connector.connect(host="localhost",
user="root", password="shraddhabondekar@101020",
                                                database="mydata2")

                my_cursor = conn.cursor()
                my_cursor.execute('select * from employee3 where '
+str(self.var_com_search.get())+" LIKE '"+str(self.search.get()+"%'"')
                rows=my_cursor.fetchall()
                if len(rows)!=0:
                    #
self.employee_table.delete(*self.employee_table.get_children())
                    for i in rows:
                        self.employee_table.insert("",END,values=i)
                conn.commit()
                conn.close()
                messagebox.showinfo('Delete', 'Employee Successfully
Deleted', parent=self.root)
            except Exception as es:
                messagebox.showerror('Error', f'Due To:{str(es)}',
parent=self.root)

if __name__=="__main__":
    root=Tk()
    obj=Employee(root)
    root.mainloop()

```



OUTPUT:

Employee Management System

EMPLOYEE MANAGEMENT SYSTEM



Employee Information

Department: Select Department

Name:

Phone No:

Designation:

Email:

Country:

Address:

Married Status: Married

Salary(CTC):

DOB:

DOJ:

ID_Proof: Select ID Proof

Gender: Male



Save

Update

Delete

Clear

Employee Table Information

Search Employee Information

Search By:

Search


Show All

Wear a Mask

Department	Name	Designation	Email	Address	Married Status	DOB	DOJ	ID Type	ID Proof	Gender	Phone	Country	Salary
HR	shraddha	dfhjjj	sfhifg@gmail.co	sfghhj	Unmarried	2000-10-10	2000-09-23		ADHAR CARD	Female	78678678	dsghijkj	23467.0
HR	shraddha	dfhjjj	sfhifg@gmail.co	sfghhj	Unmarried	2000-10-10	2000-09-23		ADHAR CARD	Female	786786789	dsghijkj	23467.0
HR	shraddha	dfhjjj	sfhifg@gmail.co	sfghhj	Unmarried	2000-10-10	2000-09-23		ADHAR CARD	Female	786786789	dsghijkj	23467.0
Select Department	gvjnn	WDGGH	gdghhhg	dhfghf	Unmarried	2000-02-21	2012-03-04	Select ID Proof	PAN CARD	Female	45686987	fghghgj	34999.0

Employee Management System

EMPLOYEE MANAGEMENT SYSTEM



Employee Information

Department: Software Engineer

Name:

Phone No:

Designation:

Email:

Country:

Address:

Married Status: Single

Salary(CTC):

DOB:

DOJ:

ID_Proof: PAN CARD

Gender: Female






Save

Update

Delete


Clear

EMPLOYEE MANAGEMENT SYSTEM

Employee Information

Department: Name: Phone No:
 Designation: Email: Country:
 Address: Married Status:
 DOB: DOJ:
 ID_Proof: Gender:



Success

Employee has been added....!!!




OK

Employee Table Information
 Search Employee Information

Search By:


Department	Name	Designation	Email	Address	Married Status	DOB	DOJ	ID Type	ID Proof	Gender	Phone	Country	Salary
HR	shraddha	dfhjij	sfhHfg@gmail.co	sfghhj	Unmarried	2000-10-10	2000-09-23		ADHAR CARD	Female	78678678	dsghijkj	23467.0
HR	shraddha	dfhjij	sfhHfg@gmail.co	sfghhj	Unmarried	2000-10-10	2000-09-23		ADHAR CARD	Female	786786789	dsghijkj	23467.0
HR	shraddha	dfhjij	sfhHfg@gmail.co	sfghhj	Unmarried	2000-10-10	2000-09-23		ADHAR CARD	Female	786786789	dsghijkj	23467.0
Select Department	gvin	WDGGH	gdghhhg	dfghf	Unmarried	2000-02-21	2012-03-04	Select ID Proof	PAN CARD	Female	45686987	fhghgj	34999.0
HR	shraddha	dfhjij	sfhHfg@gmail.co	sfghhj	Unmarried	2000-10-10	2000-09-23		ADHAR CARD	Female	78678678	dsghijkj	23467.0
HR	shraddha	dfhjij	sfhHfg@gmail.co	sfghhj	Unmarried	2000-10-10	2000-09-23		ADHAR CARD	Female	786786789	dsghijkj	23467.0

EMPLOYEE MANAGEMENT SYSTEM

Employee Information

Department: Name: Phone No:
 Designation: Email: Country:
 Address: Married Status:
 DOB: DOJ:
 ID_Proof: Gender:



Update

Are sure update this employee data

Yes No

Employee Table Information
 Search Employee Information

Search By:

Department	Name	Designation	Email	Address	Married Status	DOB	DOJ	ID Type	ID Proof	Gender	Phone	Country	Salary
HR	shraddha	dfhjij	sfhHfg@gmail.co	sfghhj	Unmarried	2000-10-10	2000-09-23		ADHAR CARD	Female	78678678	dsghijkj	23467.0
HR	shraddha	dfhjij	sfhHfg@gmail.co	sfghhj	Unmarried	2000-10-10	2000-09-23		ADHAR CARD	Female	786786789	dsghijkj	23467.0
HR	shraddha	dfhjij	sfhHfg@gmail.co	sfghhj	Unmarried	2000-10-10	2000-09-23		ADHAR CARD	Female	786786789	dsghijkj	23467.0
Select Department	gvin	WDGGH	gdghhhg	dfghf	Unmarried	2000-02-21	2012-03-04	Select ID Proof	PAN CARD	Female	45686987	fhghgj	34999.0
HR	shraddha	dfhjij	sfhHfg@gmail.co	sfghhj	Unmarried	2000-10-10	2000-09-23		ADHAR CARD	Female	78678678	dsghijkj	23467.0
HR	shraddha	dfhjij	sfhHfg@gmail.co	sfghhj	Unmarried	2000-10-10	2000-09-23		ADHAR CARD	Female	786786789	dsghijkj	23467.0

EMPLOYEE MANAGEMENT SYSTEM



Employee Information

Department: Name: Phone No.:
 Designation: Email: Country:
 Address: Married Status:
 DOB: DOJ:
 ID_Proof: Gender:



Employee Table Information

Search Employee Information

Search By:		Search	Show All	Wear a Mask									
Department	Name	Designation	Email	Address	Married Status	DOB	DOJ	ID Type	ID Proof	Gender	Phone	Country	Salary
HR	shraddha	dfhhjjj	sfhhffg@gmail.co	sfghhj	Unmarried	2000-10-10	2000-09-23		ADHAR CARD	Female	78678678	dsghijkj	23467.0
HR	shraddha	dfhhjjj	sfhhffg@gmail.co	sfghhj	Unmarried	2000-10-10	2000-09-23		ADHAR CARD	Female	786786789	dsghijkj	23467.0
HR	shraddha	dfhhjjj	sfhhffg@gmail.co	sfghhj	Unmarried	2000-10-10	2000-09-23		ADHAR CARD	Female	786786789	dsghijkj	23467.0
Select Department	gynnn	WDGGH	gdghhhg	dhfghf	Unmarried	2000-02-21	2012-03-04	Select ID Proof	PAN CARD	Female	45686987	fghghajj	34999.0
HR	shraddha	dfhhjjj	sfhhffg@gmail.co	sfghhj	Unmarried	2000-10-10	2000-09-23		ADHAR CARD	Female	78678678	dsghijkj	23467.0
HR	shraddha	dfhhjjj	sfhhffg@gmail.co	sfghhj	Unmarried	2000-10-10	2000-09-23		ADHAR CARD	Female	786786789	dsghijkj	23467.0

Employee Table Information

Search Employee Information

Search By:	Phone	Search	Show All	Wear a Mask									
Department	Name	Designation	Email	Address	Married Status	DOB	DOJ	ID Type	ID Proof	Gender	Phone	Country	Salary
HR	shraddha	dfhhjjj	sfhhffg@gmail.co	sfghhj	Unmarried	2000-10-10	2000-09-23		ADHAR CARD	Female	78678678	dsghijkj	23467.0
HR	shraddha	dfhhjjj	sfhhffg@gmail.co	sfghhj	Unmarried	2000-10-10	2000-09-23		ADHAR CARD	Female	786786789	dsghijkj	23467.0
HR	shraddha	dfhhjjj	sfhhffg@gmail.co	sfghhj	Unmarried	2000-10-10	2000-09-23		ADHAR CARD	Female	786786789	dsghijkj	23467.0
Select Department	gynnn	WDGGH	gdghhhg	dhfghf	Unmarried	2000-02-21	2012-03-04	Select ID Proof	PAN CARD	Female	45686987	fghghajj	34999.0
HR	shraddha	dfhhjjj	sfhhffg@gmail.co	sfghhj	Unmarried	2000-10-10	2000-09-23		ADHAR CARD	Female	78678678	dsghijkj	23467.0
HR	shraddha	dfhhjjj	sfhhffg@gmail.co	sfghhj	Unmarried	2000-10-10	2000-09-23		ADHAR CARD	Female	786786789	dsghijkj	23467.0