**Employees Database Management**

**Group Members:**

* Rajesh Kumar Giri (16ECE049)
* Pallavi Kumari (16ECE023)
* Avinash Prasad (16ECE026)
* Ayush Goswami (16ECE047)
* Shubham Kumar (16ECE049)

**Guided By:**

* Subhadeep Chakraborty

**Abstract:**

The “Employee Database is developed to override the problems faced during practicing manual system. This program is supported to eliminate and in some cases reduce the hardships faced by this existing manual system. Moreover this program is designed for the particular need of the company to carry out operations in a smooth and effective manner. The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this program. Thus by this all it proves it is user-friendly.

**Program Code:**

**#developed by Ayush Goswami, Shubham Kumar,Avinash Prasad, Rajesh Kumar Giri and Pallavi #Kumari**

from tkinter import \*

import sqlite3

class Project(Tk):

def \_\_init\_\_(self, \*args, \*\*kwargs):

Tk.\_\_init\_\_(self, \*args, \*\*kwargs)

container =Frame(self)

container.pack(side="top", fill="both", expand = True)

container.grid\_rowconfigure(0, weight=1)

container.grid\_columnconfigure(0, weight=1)

self.frames = {}

for F in (LogInPage, main):#, PageTwo):

frame = F(container, self)

self.frames[F] = frame

frame.grid(row=0, column=0, sticky="nsew")

self.show\_frame(LogInPage)

def show\_frame(self, cont):

frame = self.frames[cont]

frame.tkraise()

class LogInPage(Frame):

def \_\_init\_\_(self, parent, controller):

Frame.\_\_init\_\_(self,parent)

conn = sqlite3.connect('my\_employee\_database.db')

curs = conn.cursor()

def validate():

if t1.get()==up[0] and t2.get()==up[1]:

btn1=Button(self,text="Log In",font=('bold',15),bg='grey',command=lambda: controller.show\_frame(main)).place(x=340,y=300)

l1=Label(self,text=" Verified User! ",font=15,fg='red',bg='grey')

l1.place(x=270,y=270)

else:

l1=Label(self,text='Wrong username or Password!!',font=15,fg='red',bg='grey').place(x=280,y=270)

def clear():

t1.delete(0,END)

t2.delete(0,END)

up=['user','12345']

self.configure(background='grey')

Label(self,text='AnaCoders',font=('Times New Roman bold',30),fg='green2',bg='grey').place(x=300,y=30)

l1=Label(self,text='Log In',font=('bold',30),bg='grey')

l1.place(x=320,y=100)

l2=Label(self,text='Username',font=20,bg='grey').place(x=200,y=190)

l3=Label(self,text="Password",font=20,bg='grey').place(x=200,y=240)

t1=Entry(self,width=30)

t1.place(x=300,y=190)

t2=Entry(self,width=30,show='\*\*')

t2.place(x=300,y=240)

btn=Button(self,text="Verify",font=('bold',15),bg='grey',command=validate).place(x=340,y=300)

btn1=Button(self,text="Refresh",font=10,bg='grey',command=clear).place(x=500,y=210)

class main(Frame):

def \_\_init\_\_(self, parent, controller):

Frame.\_\_init\_\_(self, parent)

label =Label(self, text="Employees DataBase",bg='grey', font=('bold',40))

label.pack(pady=10,padx=10)

conn = sqlite3.connect('my\_employee\_database.db')

curs = conn.cursor()

def submit():

lb.delete(0,END)

insert\_command = """INSERT OR IGNORE INTO employee4(first\_name,last\_name,employee\_id,gender,designation) VALUES('%s', '%s', '%s', '%s', '%s');"""

if len(e1.get())!=0 and len(e2.get())!=0 and len(e3.get())!=0 and len(e4.get())!=0 and len(e5.get())!=0 :

curs.execute(insert\_command%(e1.get(),e2.get(),e3.get(),e4.get(),e5.get()))

conn.commit()

lb.insert('END','Successfully Added')

else:

lb.insert(END,"Empty fields")

def view():

lb.delete(0,END)

curs.execute('select \* from employee4 ')

result=curs.fetchall()

r=0

c=290

for row in result:

name=row[0]+" "+row[1]

print(name)

for rows in range (len(result)):

s=[]

for cols in range (5):

s.append(result[rows][cols])

lb.insert(r,s)

r+=1

def search():

lb.delete(0,END)

if len(e3.get())!=0:

curs.execute("select \* from employee4 where employee\_id=?",(e3.get(),))

result=curs.fetchall()

print(len(result))

r=0

c=290

for rows in range (len(result)):

s=[]

print(len(result))

for cols in range (5):

s.append(result[rows][cols])

lb.insert(r,s)

r+=1

def delet():

lb.delete(0,END)

if len(e3.get())!=0:

curs.execute("delete from employee4 where employee\_id=?",(e3.get(),))

result=curs.fetchall()

print(len(result))

lb.delete(END,"ROw deleted")

conn.commit()

def update():

lb.delete(0,END)

if len(e3.get())!=0:

print("Hel")

if len(e1.get())!=0:

curs.execute("update employee4 set first\_name=? where employee\_id=?",(e1.get(),e3.get(),))

conn.commit()

if len(e2.get())!=0:

curs.execute("update employee4 set last\_name=? where employee\_id=?",(e2.get(),e3.get(),))

conn.commit()

if len(e4.get())!=0:

curs.execute("update employee4 set gender=? where employee\_id=?",(e4.get(),e3.get(),))

conn.commit()

if len(e5.get())!=0:

curs.execute("update employee4 set designation=? where employee\_id=?",(e5.get(),e3.get(),))

conn.commit()

if len(e1.get())!=0 or len(e2.get())!=0 or len(e4.get())!=0 or len(e5.get())!=0:

lb.insert(END,'Updated')

else:

lb.insert(END,'No Data Given!!!!!')

def clear():

lb.delete(0,END)

e1.delete(0,END)

e2.delete(0,END)

e3.delete(0,END)

e4.delete(0,END)

e5.delete(0,END)

self.config(background='grey')

Label(self, text="First Name",bg='grey',font=20).place(x=50,y=100)

Label(self, text="Last Name",bg='grey',font=20).place(x=50,y=130)

Label(self, text="Employee ID",bg='grey',font=20).place(x=50,y=160)

Label(self, text="Gender",font=20,bg='grey').place(x=50,y=190)

Label(self, text="Designation",bg='grey',font=20).place(x=50,y=220)

e1 = Entry(self,width=30)

e1.place(x=150,y=100)

e2 = Entry(self,width=30)

e2.place(x=150,y=130)

e3 = Entry(self,width=30)

e3.place(x=150,y=160)

e4 = Entry(self,width=30)

e4.place(x=150,y=190)

e5 = Entry(self,width=30)

e5.place(x=150,y=220)

lb=Listbox(self,height=13,width=125)

lb.place(x=12,y=300)#list box

Button(self, text='Search',bg='grey',font=('bold',17),command=search).place(x=370,y=152)

Button(self, text='View',bg='grey',font=20,width=6,command=view).place(x=212,y=267)

Button(self, text="Log out",bg='grey',font=('bold',15),fg='blue',command=lambda: controller.show\_frame(LogInPage)).place(x=680,y=80)

Button(self,text='Clear',font=20,bg='grey',width=6,command=clear).place(x=276,y=267)

Button(self, text='Add',font=20,bg='grey',width=6,command=submit).place(x=20,y=267)

Button(self, text='Update',bg='grey',font=20,width=6,command=update).place(x=84,y=267)

Button(self, text='Delete',font=20,bg='grey',width=6,command=delet).place(x=148,y=267)

sb=Scrollbar(self)

sb.place(x=765,y=390)

app = Project()

app.title("Employees DataBase")

app.geometry('780x520')

app.mainloop()

Output:



