**AirLink InfoTech**

***-* *Soar to your***

***destination with smiles***

***packed into your luggage***

**Adithyan Pratheeksh Nair**

**&**

**Ryan Jomy Valavi**

**DESCRIPTION OF THE PROJECT**

**Aim:**

**The project aims to mimic the working of a database management software of a particular company.**

**Objective:**

**To obtain data from the user and perform several functions with it using a well-defined set of algorithms.**

**Description:**

**Employee and sales data are inputted into MySQL tables. These records can be updated, deleted, analyzed and compared with the other records in the table.**

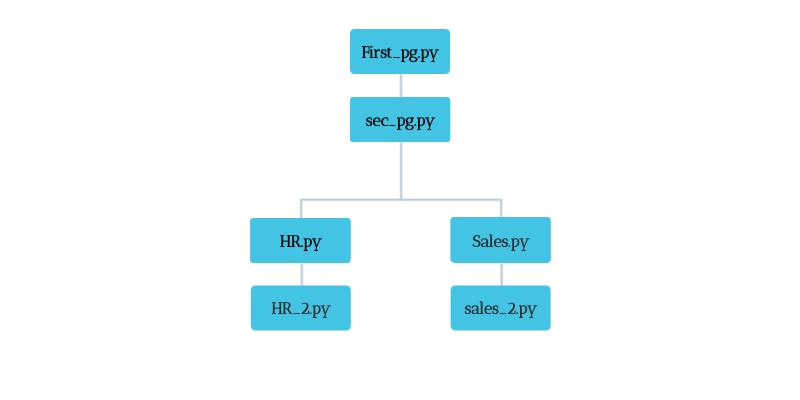
**The essence of the project is the process through which a company by the name of Airlink Infotech maintains its database of records containing the information of its employees which is optimized into a MySQL table. This is a ticketing agency that books tickets for its customers.**

**The code is designed in such a way that it accommodates two sections of this company namely the sales and the HR department. It gives complete freedom to all the employees falling under the HR department to modify, delete and update data pertaining to all the employees.**

**Another table is created containing the data of all the transactions made through this firm. This can only be accessed by the sales department of this firm. The data from this is used to compare the sales of different flights, modify the data if needed, track particular transactions and graphically the analyze the growth of various flights through the course of the year.**

**Thus, this product can be used by the company to provide a systematic approach to sort and alter the bulk of data a company like a ticketing agency receives on a daily basis.**

**Program Flowchart:**

****

**CONCEPTS USED:**

* **MySQL**
* **Python-MySQL connectivity**
* **Iterations and Conditional statements**
* **Data Structures-Lists, Dictionaries**
* **User Interface Creation-Tkinter**
* **Python Modules and Libraries**

**#First\_pg.py**

from tkinter import \*

from tkinter import messagebox

#CREATING TKINTER WINDOW

main=Tk()

main.state('zoomed')

main.configure(bg='cyan')

b1=Button(main,text='AirLink InfoTech',bd='0',bg='cyan',font=('Times Roman Bold',60))

b1.pack(pady=100)

#COMMAND FOR NEXT PAGE

def \_1():

main.destroy()

import sec\_pg #IMPORTING sec\_pg.py

#COMMAND TO QUIT THE PROGRAM

def quit\_pg():

mb=messagebox.askquestion('quit','Are you sure that you want to quit')

if mb=='yes':

main.destroy()

#CREATING BUTTONS

b2=Button(main,text='NEXT',font=('',30),command=\_1,relief='solid')

b2.place(relx=0.85,rely=0.8)

b3=Button(main,text='QUIT',font=('',30),command=quit\_pg,relief='solid')

b3.place(rely=0.8,relx=0.03)



**#sec\_pg.py**

from tkinter import \*

#COMMAND FOR HR PAGE

def hr\_pg():

m.destroy()

import HR #IMPORTING HR.py

#COMMAND FOR SALES PAGE

def sales\_pg():

m.destroy()

import Sales #IMPORTING Sales.py

#CREATING NEW WINDOW

m=Tk()

m.state('zoomed')

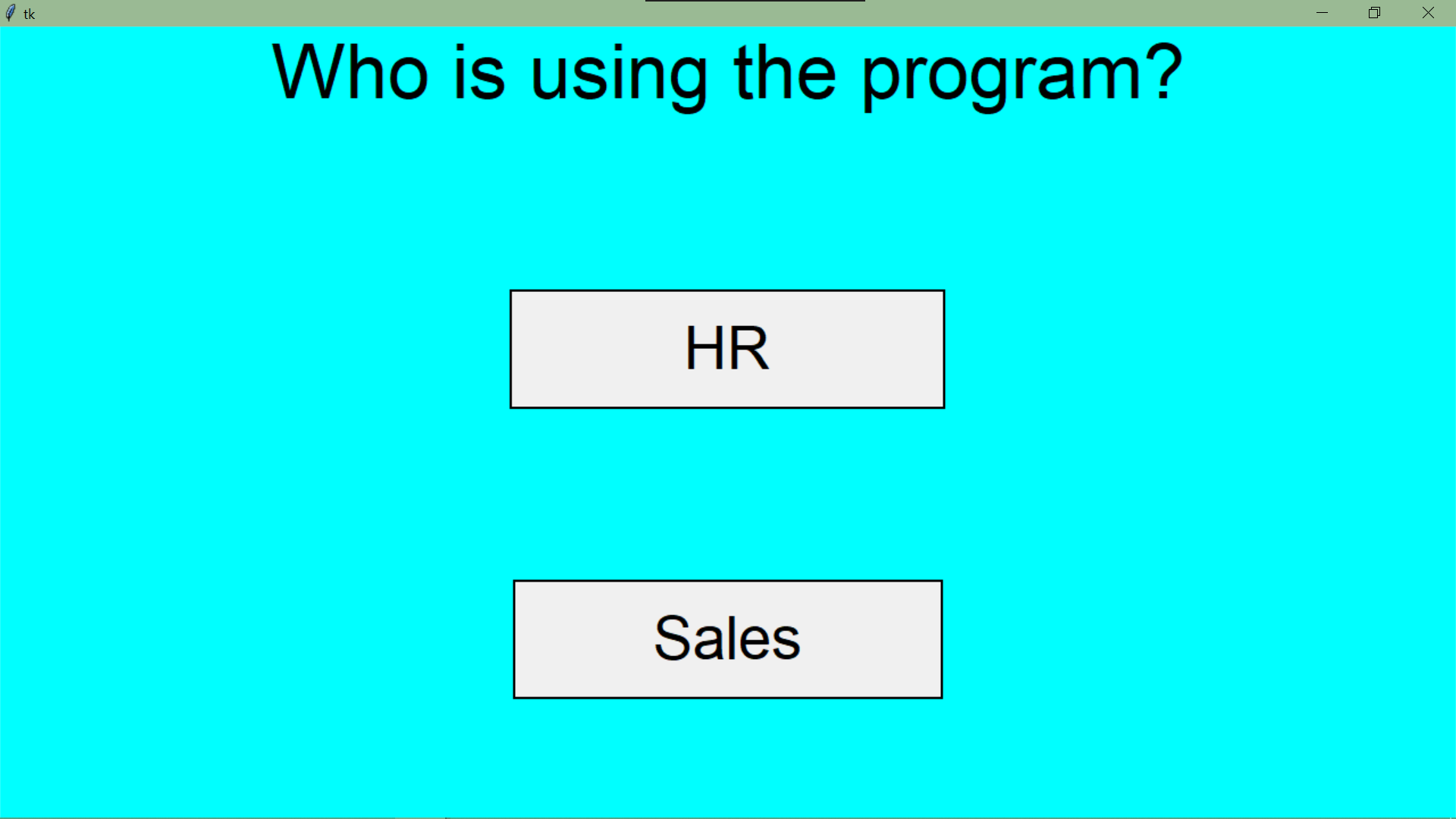
m.configure(bg='cyan')

Label(m,bg='cyan',text='Who is using the program?',font=('',50)).pack()

Button(m,text="HR",command=hr\_pg,font=('',40),relief='solid').pack(ipadx=130,pady=150)

Button(m, text="Sales",command=sales\_pg,font=('',40),relief='solid').pack(ipadx=100)

m.mainloop()



**#HR.py**

from tkinter import \*

from tkinter import messagebox,ttk

import mysql.connector as c

hr=Tk()

hr.state('zoomed')

hr.config(bg='cyan')

con=c.connect(host='localhost',user='root',passwd='achuadivava@1438',database='airinfo')

x=con.cursor()

#back

def back():

hr.destroy()

import First\_pg

#Username

user\_label=Label(hr,text='Username',bg='cyan',font=('',25))

user\_label.pack(pady=30)

user=Entry(hr,font=('',30))

user.pack(padx=50)

#password

pass\_label=Label(hr,text='Password',bg='cyan',font=('',25))

pass\_label.pack(pady=30)

passw=Entry(hr,font=('',30),show='\*')

passw.pack()

#next page

def check():

n=user.get() #USERNAME

p=passw.get() #PASSWORD

if n=='': #IF USERNAME IS NOT ENTERED

messagebox.showerror('Error','Invalid username')

elif p=='': #IF PASSWORD IS NOT ENTERED

messagebox.showerror('Error','Invalid password')

else:

query='select \* from hrpass where usern="{}" and passw="{}"'.format(n,p)

x.execute(query)

y=x.fetchone()

if y==None: #IF USERNAME OR PASSWORD DOESN'T EXIST IS TABLE

messagebox.showerror('Error','Incorrect Username or Password')

else:

user.destroy(),user\_label.destroy(),pass\_label.destroy()

passw.destroy(),submit\_btn.destroy(),back\_btn.destroy()

def no(): #COMMAND FOR NOT USING FOR THE FIRST TIME

hr.destroy()

import HR\_2 #IMPORTING HR\_2.py

def yes(): #COMMAND FOR USING FOR THE FIRST TIME

l1.destroy(),l2.destroy(),yes\_btn.destroy(),no\_btn.destroy()

def nxt():

e=emp\_no.get() #EMPLOYEE ID

n=nm.get() #NAME

d=(dept.get()).lower() #DEPARTMENT

s=sal.get() #SALARY

j="select emp\_no from employee where emp\_no='{}'".format(e)

x.execute(j)

k=x.fetchone()

if k!=None or e=='': #IF EMPLOYEE ID IS NOT ENTERED OR ALREADY EXIST IN EMPLOYEE TABLE

messagebox.showerror('Error','Invalid Employee No')

elif n=='':

messagebox.showerror('Error','Invalid Employee Name')

elif s=='':

messagebox.showerror('Error','Salary not entered')

elif d not in ('hr','sales'):

messagebox.showerror('Error','Invalid Department')

else:

query1="insert into employee values('{}','{}','{}',{})".format(e,n,d,int(s))

x.execute(query1)

con.commit()

def add():

u=usern.get()

p=passw.get()

if u=='':

messagebox.showerror('Error','Username not entered')

elif p=='':

messagebox.showerror('Error','Password not entered')

else:

hr.destroy()

if d=="hr":

query2="insert into hrpass values('{}','{}','{}')".format(e,u,p)

x.execute(query2)

con.commit()

import HR\_2

elif d=='sales':

query3="insert into salespass values('{}','{}','{}')".format(e,u,p)

x.execute(query3)

con.commit()

import HR\_2

empno\_label.destroy(),nm\_label.destroy(),dept\_label.destroy(),sal\_label.destroy()

emp\_no.destroy(),nm.destroy(),dept.destroy(),sal.destroy(),btn.destroy()

usern\_label=Label(hr,text='Username for the employee',bg='cyan',font=('',40))

usern\_label.place(relx=0.025,rely=0.05)

usern=Entry(hr,font=('',30))

usern.pack(anchor='e',padx=100,pady=40)

passw\_label=Label(hr,text='Password for the employee',bg='cyan',font=('',40))

passw\_label.place(relx=0.025,rely=0.24)

passw=Entry(hr,font=('',30),show='•')

passw.pack(anchor='e',padx=100,pady=40)

Button(hr,text='Next',font=('',40),command=add,relief='solid').pack()

empno\_label=Label(hr,text='Employee id',bg='cyan',font=('',40))

empno\_label.place(relx=0.025,rely=0.05)

emp\_no=Entry(hr,font=('',30))

emp\_no.pack(anchor='e',padx=100,pady=40)

nm\_label=Label(hr,text='Name of Employee',bg='cyan',font=('',40))

nm\_label.place(relx=0.025,rely=0.24)

nm=Entry(hr,font=('',30))

nm.pack(anchor='e',padx=100,pady=40)

dept\_label=Label(hr,text='Department',bg='cyan',font=('',40))

dept\_label.place(relx=0.025,rely=0.65)

sal\_label=Label(hr,text='Salary',bg='cyan',font=('',40))

sal\_label.place(relx=0.025,rely=0.45)

sal=Entry(hr,font=('',30))

sal.pack(anchor='e',padx=100,pady=40)

l=['HR','Sales']

txt=StringVar()

dept=ttk.Combobox(hr,textvariable=txt,font=('',30))

dept['values']=l

dept.current(0)

dept.pack(anchor='e',padx=100,pady=40)

btn=Button(hr,text='Continue',font=('',40),command=nxt,relief='solid')

btn.pack()

l1=Label(hr,text='Welcome to the\nHR department of Airlink Infotech',bg='cyan',font=('',45))

l1.pack()

l2=Label(hr,text='Are you using the program for the first time?',bg='cyan',font=('',40))

l2.pack(pady=100)

yes\_btn=Radiobutton(hr,text='Yes',font=('',40),command=yes,relief='solid')

yes\_btn.pack(side='right')

no\_btn=Radiobutton(hr,text='No',font=('',40),command=no,relief='solid')

no\_btn.pack(side='left')

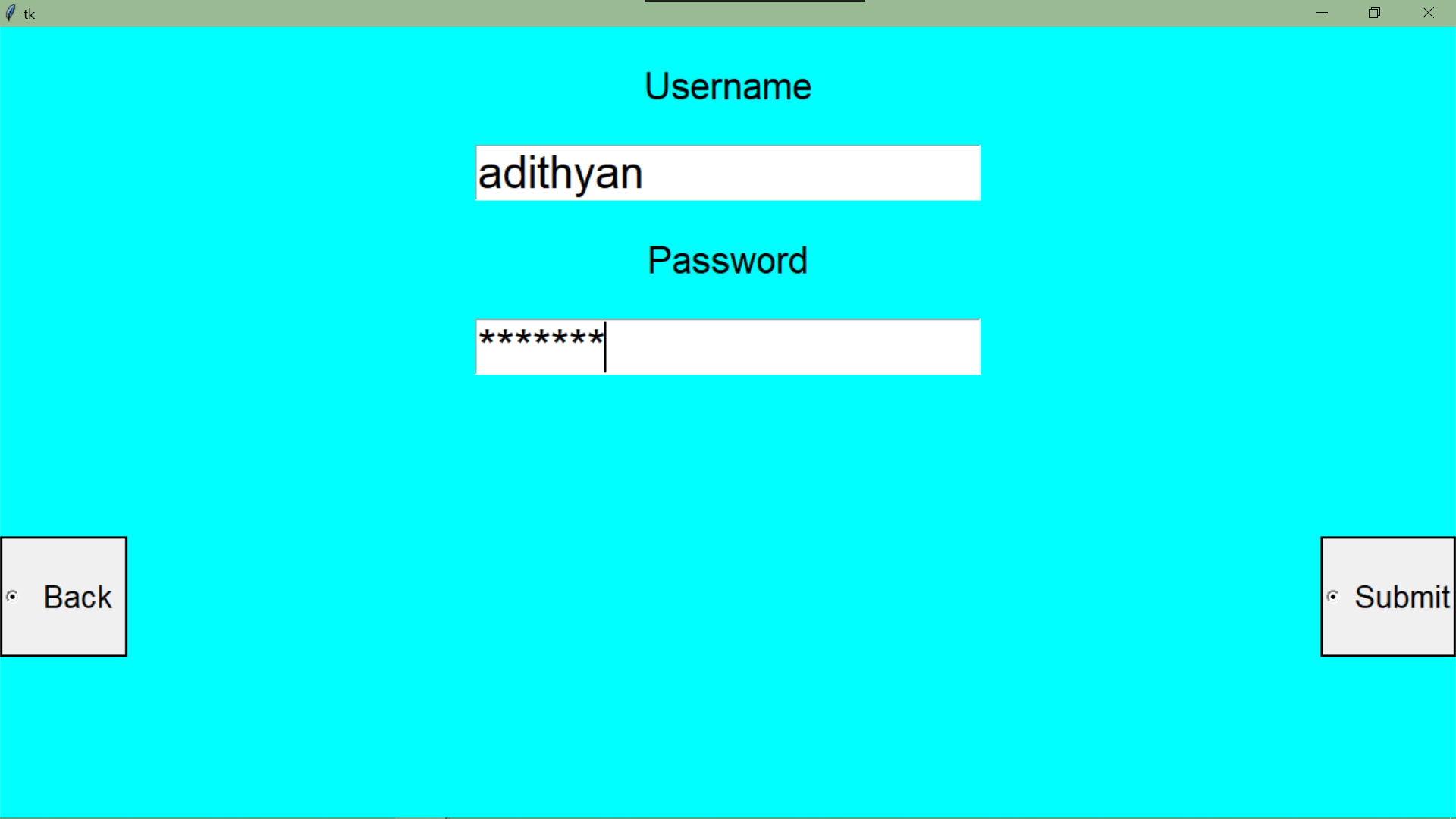
submit\_btn=Radiobutton(hr,text='Submit',font=('',20),command=check,relief='solid')

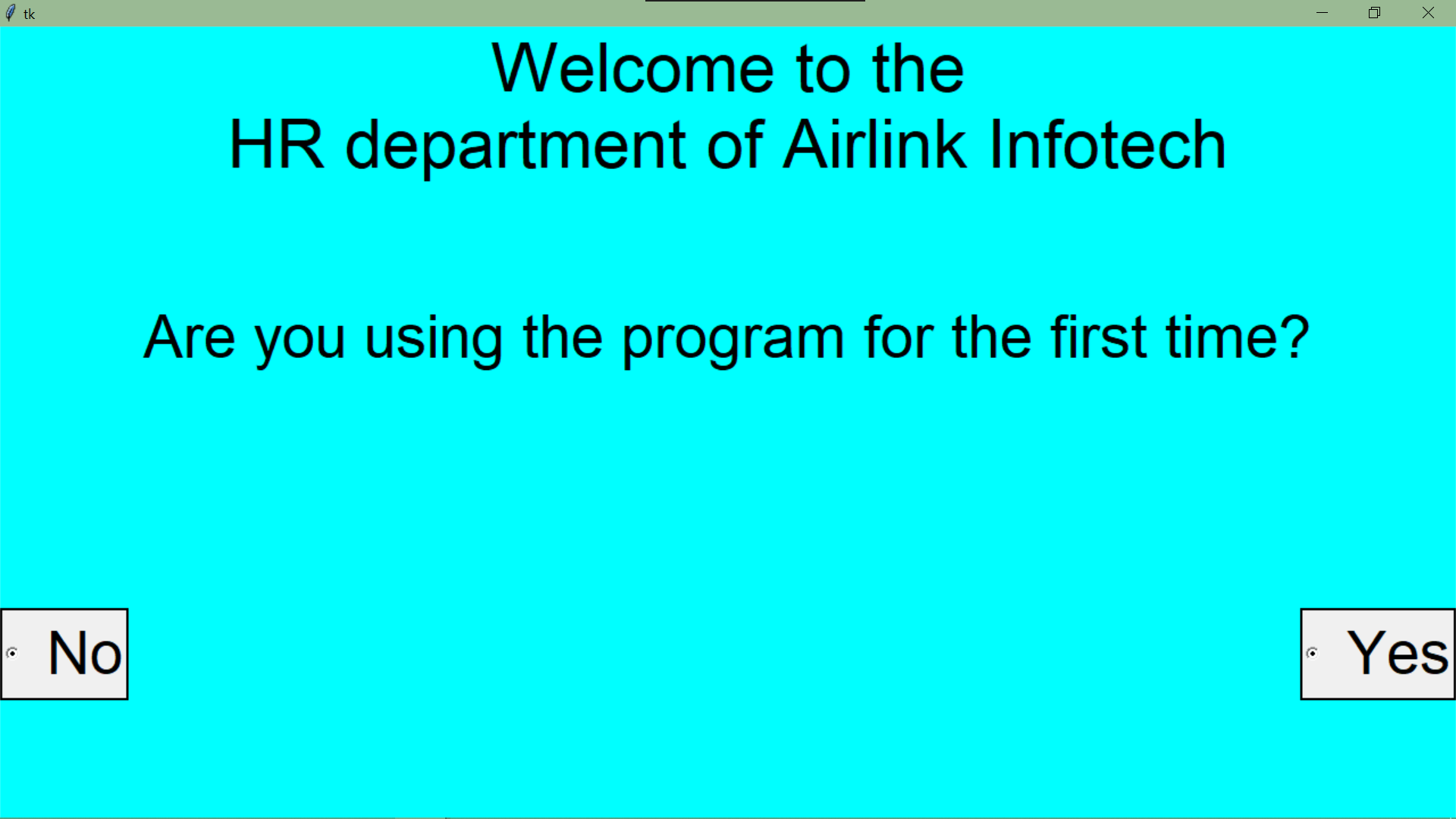
submit\_btn.pack(side='right',pady=10,ipady=30)

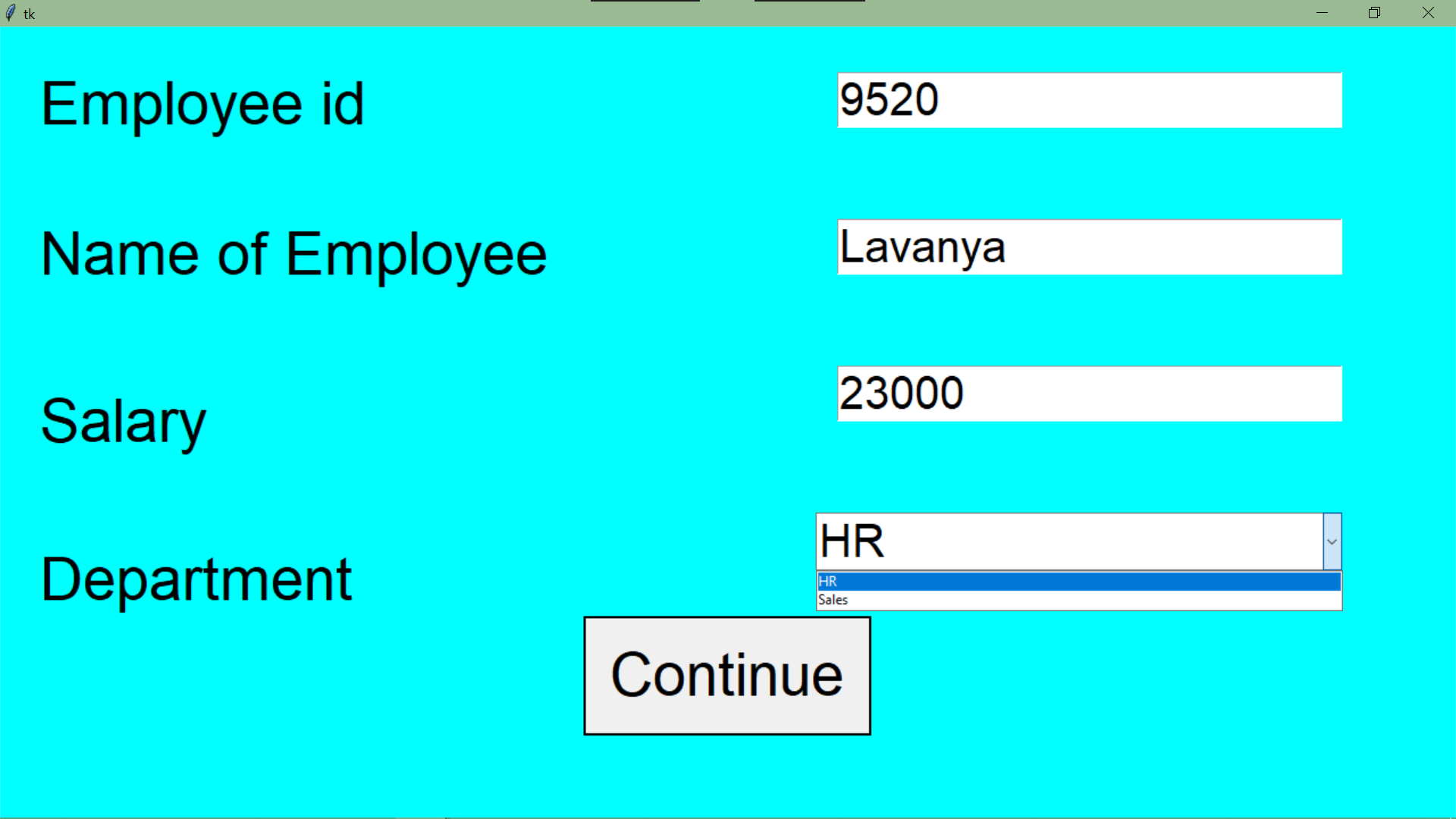
back\_btn=Radiobutton(hr,text=' Back ',font=('',20),command=back,relief='solid')

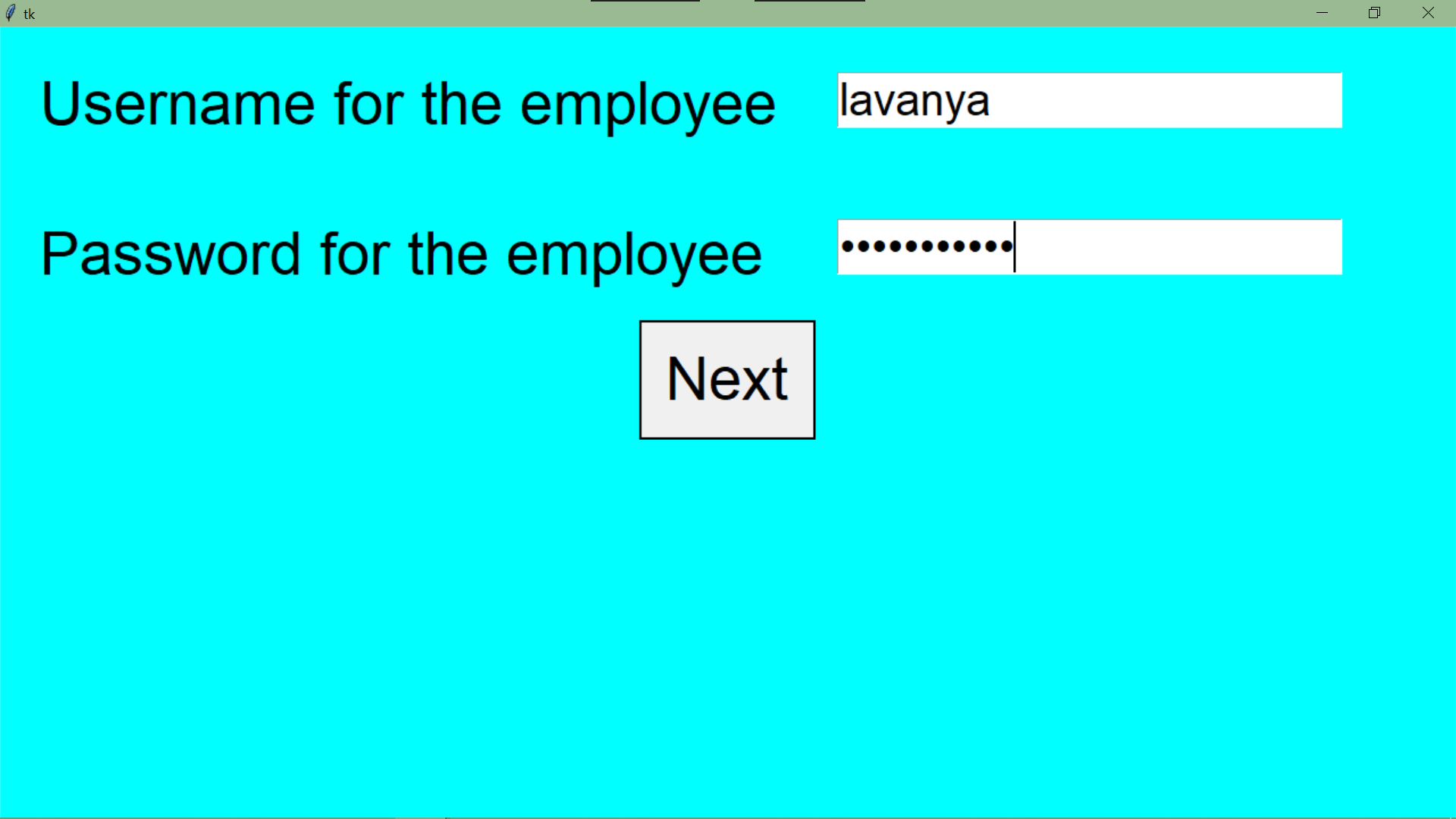
back\_btn.pack(side='left',pady=10,ipady=30)

hr.mainloop()









**#HR\_2.py**

from tkinter import \*

from tkinter import ttk,messagebox

import mysql.connector as c

#CONNECTING MYSQL

con=c.connect(host='localhost',user='root',passwd='achuadivava@1438',database='airinfo')

x=con.cursor()

#CREATING NEW WINDOW

HR=Tk()

HR.state('zoomed')

HR.config(bg='cyan')

#COMMAND TO GO BACK

def home():

HR.destroy()

import sec\_pg #IMPORTING sec\_pg.py

def nxt():

ch=lb.index(lb.curselection())

def bck():

hr.destroy()

HR.state('zoomed')

#TO ADD AN EMPLOYEE

if ch==0:

def nxt():

def submit():

u=usern.get() #USERNAME

p=passw.get() #PASSWORD

if u=='': #IF NOTHING IS ENTERED

messagebox.showerror('Error','Username not entered')

elif p=='':

messagebox.showerror('Error','Password not entered')

else:

if dpt=="hr":

query2="insert into hrpass values('{}','{}','{}')".format(emp\_id,u,p)

x.execute(query2)

con.commit() #INSERTING THE DATA INTO HRPASS TABLE

elif dpt=='sales':

query3="insert into salespass values('{}','{}','{}')".format(emp\_id,u,p)

x.execute(query3)

con.commit() #INSERTING THE DATA INTO SALESPASS TABLE

mb=messagebox.showinfo('','Added successfully')

emp\_id=empid.get() #EMPLOYEE ID

emp\_nm=empnm.get() #EMPLOYEE NAME

dpt=(dept.get()).lower() #DEPARTMENT

sly=sal.get() #SALARY

j="select emp\_no from employee where emp\_no='{}'".format(emp\_id)

x.execute(j)

k=x.fetchone()

if k!=None or emp\_id=='': #IF EMPLOYEE ID IS NOT ENTERED OR ALREADY EXIST IN EMPLOYEE TABLE

messagebox.showerror('Error','Invalid Employee ID')

elif emp\_nm=='': #IF EMPLOYEE NAME IS NOT ENTERED

messagebox.showerror('Error','Invalid Employee Name')

elif sly=='': #IF SALARY IS NOT ENTERED

messagebox.showerror('Error','Invalid Salary')

elif dpt not in ('hr','sales'): #IF DEPARTMENT ENTERED IS NEITHER HR NOR SALES

messagebox.showerror('Error','Invalid Department')

else:

query1="insert into employee values('{}','{}','{}',{})".format(emp\_id,emp\_nm,dpt,int(sly))

x.execute(query1)

con.commit()

LABEL.destroy(),empid\_label.destroy(),empnm\_label.destroy(),dept\_label.destroy(),sal\_label.destroy()

empid.destroy(),empnm.destroy(),dept.destroy(),sal.destroy(),nxt.destroy()

usern\_label=Label(hr,text='Username to be assigned',bg='cyan',font=('',25))

usern\_label.place(relx=0.025,rely=0.15)

usern=Entry(hr,font=('',22))

usern.pack(anchor='ne',padx=20,pady=95)

passw\_label=Label(hr,text='Password to be assigned',bg='cyan',font=('Times Roman',25))

passw\_label.place(relx=0.027,rely=0.34)

passw=Entry(hr,font=('',22),show='\*')

passw.pack(anchor='ne',padx=20)

add=Button(hr,text='Add',font=('',40),command=submit,relief='solid')

add.place(relx=0.85,rely=0.8)

HR.state('withdraw')

hr=Tk()

hr.state('zoomed')

hr.config(bg='cyan')

LABEL=Label(hr,text='TO ADD AN EMPLOYEE',bg='cyan',font=('',40))

LABEL.place(x=0,y=0,relwidth=1)

empid\_label=Label(hr,text='Employee Id',bg='cyan',font=('',25))

empid\_label.place(relx=0.025,rely=0.15)

empid=Entry(hr,font=('',30))

empid.pack(anchor='nw',padx=350,pady=95)

empnm\_label=Label(hr,text='Name of employee',bg='cyan',font=('',25))

empnm\_label.place(relx=0.027,rely=0.34)

empnm=Entry(hr,font=('',30))

empnm.pack(anchor='nw',padx=350)

sal\_label=Label(hr,text='Salary',bg='cyan',font=('',25))

sal\_label.place(relx=0.0325,rely=0.55)

sal=Entry(hr,font=('',30))

sal.pack(anchor='nw',padx=350,pady=95)

dept\_label=Label(hr,text='Department',bg='cyan',font=('',25))

dept\_label.place(relx=0.04,rely=0.775)

l=['HR','Sales']

txt=StringVar()

dept=ttk.Combobox(hr,textvariable=txt,font=('',30))

dept['values']=l

dept.current(0)

dept.pack(anchor='nw',padx=350)

nxt=Button(hr,text='Next',font=('',40),command=nxt,relief='solid')

nxt.place(relx=0.85,rely=0.85)

back=Button(hr,text='Back',font=('',40),command=bck,relief='solid')

back.place(relx=0.02,rely=0.85)

#Deleting an existing employee

if ch==1:

def submit():

emp\_id=empid.get()

dpt=(dept.get()).lower()

j="select emp\_no from employee where emp\_no='{}'".format(emp\_id)

x.execute(j)

k=x.fetchone()

if k==None or emp\_id=='': #IF EMPLOYEE ID IS NOT ENTERED OR ALREADY EXIST IN EMPLOYEE TABLE

messagebox.showerror('Error','Invalid Employee ID')

elif dpt not in ('hr','sales'):

messagebox.showerror('Error','Invalid Department')

else:

if dpt=="hr":

query="delete from hrpass where emp\_no='{}'".format(emp\_id)

x.execute(query)

con.commit()

query1="delete from employee where emp\_no='{}'".format(emp\_id)

x.execute(query1)

con.commit()

mb=messagebox.showinfo('','Deleted successfully')

elif dpt=='sales':

query="delete from salespass where emp\_no='{}'".format(emp\_id)

x.execute(query)

con.commit()

query1="delete from employee where emp\_no='{}'".format(emp\_id)

x.execute(query1)

con.commit()

mb=messagebox.showinfo('','Deleted successfully')

HR.state('withdraw')

hr=Tk()

hr.state('zoomed')

hr.config(bg='cyan')

LABEL=Label(hr,text='TO DELETE AN EMPLOYEE',bg='cyan',font=('Times Roman',40))

LABEL.place(x=0,y=0,relwidth=1)

empid\_label=Label(hr,text='Employee Id',bg='cyan',font=('Times Roman',25))

empid\_label.place(relx=0.025,rely=0.15)

empid=Entry(hr,font=('',30))

empid.pack(anchor='nw',padx=350,pady=95)

dept\_label=Label(hr,text='Department',bg='cyan',font=('Times Roman',25))

dept\_label.place(relx=0.0325,rely=0.5)

l=['HR','Sales']

txt=StringVar()

dept=ttk.Combobox(hr,textvariable=txt,font=('',30))

dept['values']=l

dept.current(0)

dept.pack(anchor='nw',padx=350,pady=95)

delete=Button(hr,text='DELETE',font=('',40),command=submit,relief='solid')

delete.place(relx=0.8,rely=0.8)

back=Button(hr,text='Back',font=('',40),command=bck,relief='solid')

back.place(relx=0.02,rely=0.8)

#Displaying the employee database

if ch==2:

hr=Tk()

hr.state('zoomed')

L=Label(hr,text='Details of an Employee',font=('',40))

L.place(relwidth=1)

L1=Label(hr,text='Employee ID',font=('',40),borderwidth=2,relief='solid')

L1.place(rely=0.1,relx=0.01,relwidth=0.24)

L2=Label(hr,text='Name',font=('',40),borderwidth=2,relief='solid')

L2.place(rely=0.1,relx=0.251,relwidth=0.3)

L3=Label(hr,text='Department',font=('',40),borderwidth=2,relief='solid')

L3.place(rely=0.1,relx=0.5515,relwidth=0.27)

L4=Label(hr,text='Salary',font=('',40),borderwidth=2,relief='solid')

L4.place(rely=0.1,relx=0.823,relwidth=0.165)

query="select \* from employee order by name"

x.execute(query)

j=0.2

for k in x:

l1=Label(hr,text=k[0],font=('',30),borderwidth=2,relief='solid')

l1.place(rely=j,relx=0.01,relwidth=0.24)

l2=Label(hr,text=k[1],font=('',30),borderwidth=2,relief='solid')

l2.place(rely=j,relx=0.251,relwidth=0.3)

l3=Label(hr,text=k[2],font=('',30),borderwidth=2,relief='solid')

l3.place(rely=j,relx=0.5515,relwidth=0.27)

l4=Label(hr,text=k[3],font=('',30),borderwidth=2,relief='solid')

l4.place(rely=j,relx=0.823,relwidth=0.165)

j+=0.08

#Updating the salary package of a particular employee

if ch==3:

def submit():

emp\_id=empid.get()

sly=sal.get()

j="select emp\_no from employee where emp\_no='{}'".format(emp\_id)

x.execute(j)

k=x.fetchone()

if emp\_id=='' or k==None: #IF EMPLOYEE ID IS NOT ENTERED OR ALREADY EXIST IN EMPLOYEE TABLE

messagebox.showerror('Error','Invalid Employee ID')

elif sly=='':

messagebox.showerror('Error','Invalid Salary')

else:

query="update employee set Salary={} where emp\_no='{}'".format(int(sly),emp\_id)

x.execute(query)

con.commit()

mb=messagebox.showinfo('','Upadated successfully')

HR.state('withdraw')

hr=Tk()

hr.state('zoomed')

hr.config(bg='cyan')

LABEL=Label(hr,text='TO UPDATE SALARY OF AN EMPLOYEE',bg='cyan',font=('Times Roman',40))

LABEL.place(x=0,y=0,relwidth=1)

empid\_label=Label(hr,text='Employee Id',bg='cyan',font=('',25))

empid\_label.place(relx=0.025,rely=0.15)

empid=Entry(hr,font=('',30))

empid.pack(anchor='nw',padx=350,pady=95)

sal\_label=Label(hr,text='New Salary',bg='cyan',font=('',25))

sal\_label.place(relx=0.04,rely=0.35)

sal=Entry(hr,font=('',30))

sal.pack(anchor='nw',padx=350)

Button(hr,text='UPDATE',font=('',40),command=submit,relief='solid').place(relx=0.75,rely=0.8)

back=Button(hr,text='Back',font=('',40),command=back,relief='solid')

back.place(relx=0.02,rely=0.8)

#MAIN

label=Label(HR,text='Which function would you like to perform?',bg='cyan',font=('',40))

label.pack()

lb=Listbox(HR,font=('',40),relief='solid')

lb.place(rely=0.2,relx=0.25,relheight=0.4)

lb.insert(0,' Add an employee')

lb.insert(1,' Remove an employee')

lb.insert(3," Update salary package")

lb.insert(2,' Display all employees')

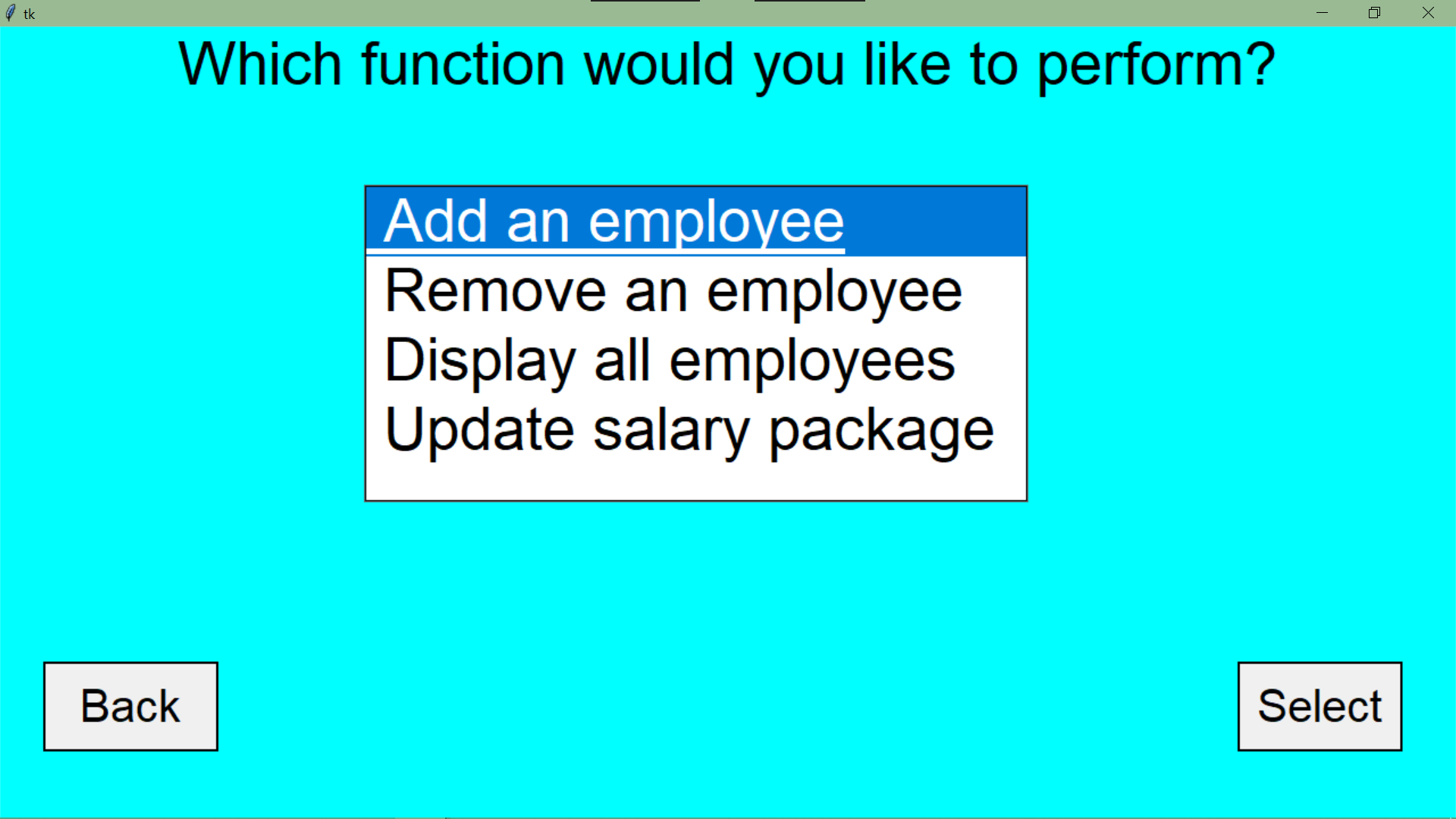
slc=Button(HR,text='Select',font=('',30),command=nxt,relief='solid')

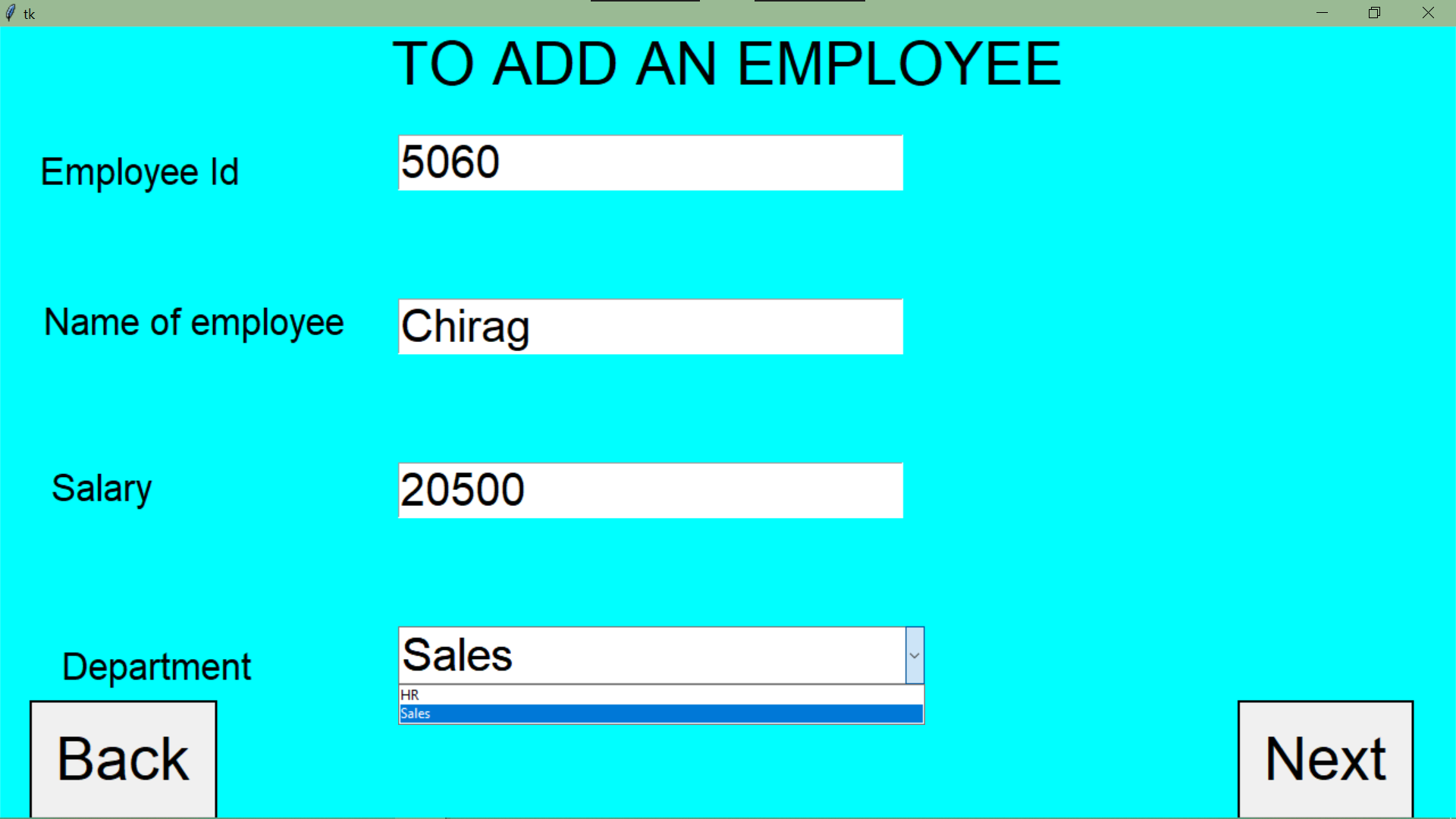
slc.place(relx=0.85,rely=0.8)

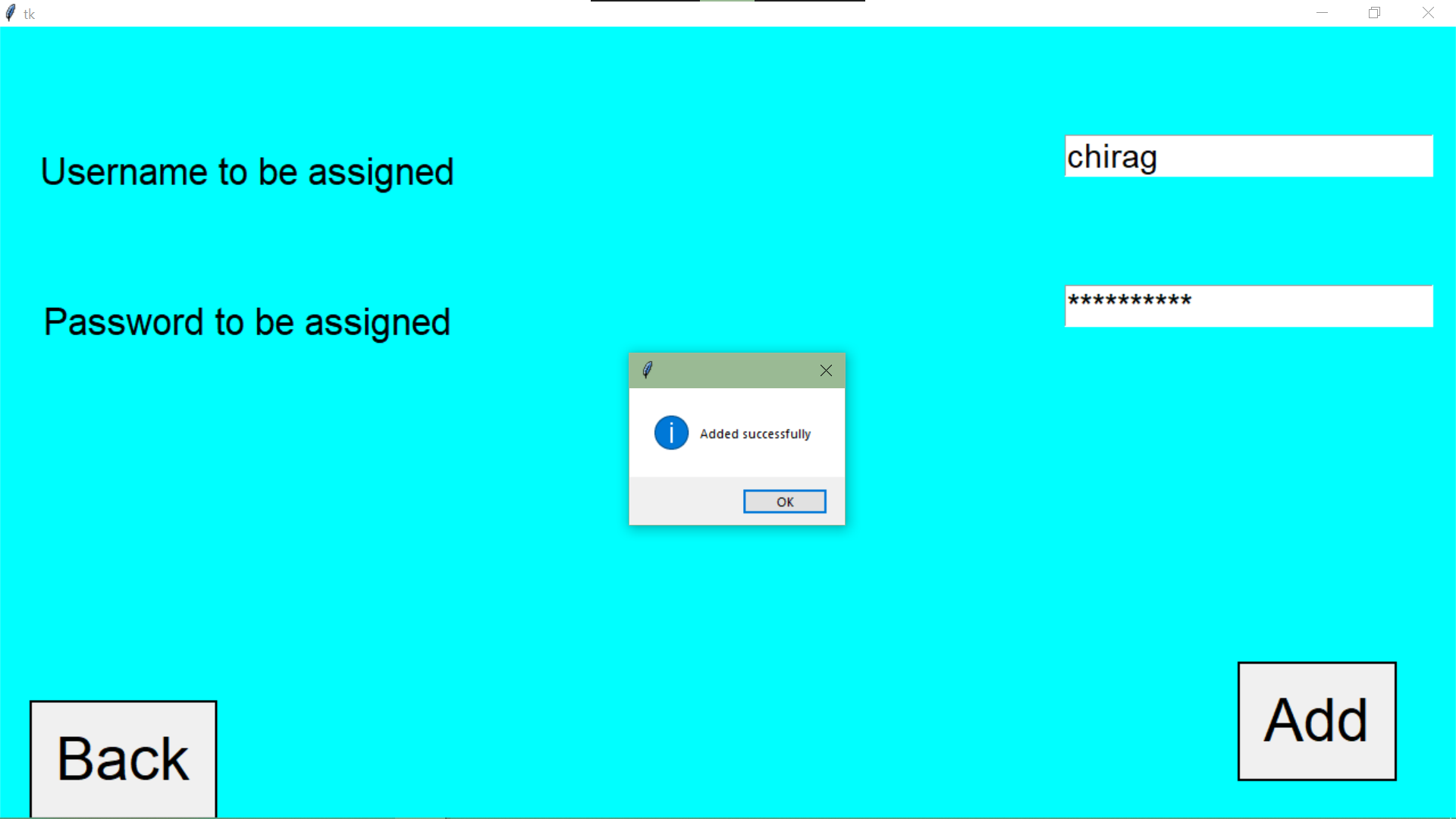
back=Button(HR,text='Back',font=('',30),command=home,relief='solid')

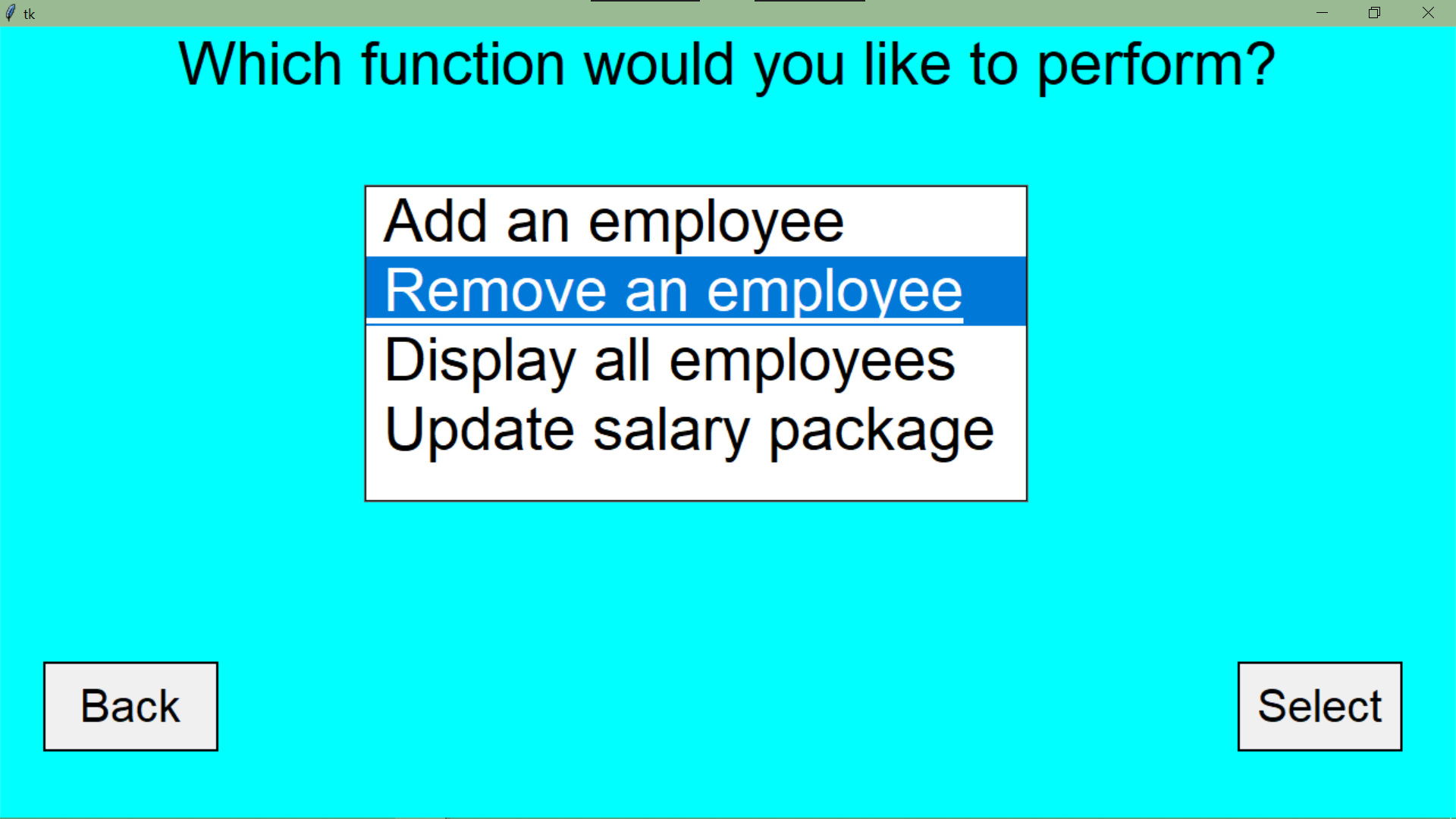
back.place(rely=0.8,relx=0.03,relwidth=0.12)

HR.mainloop()

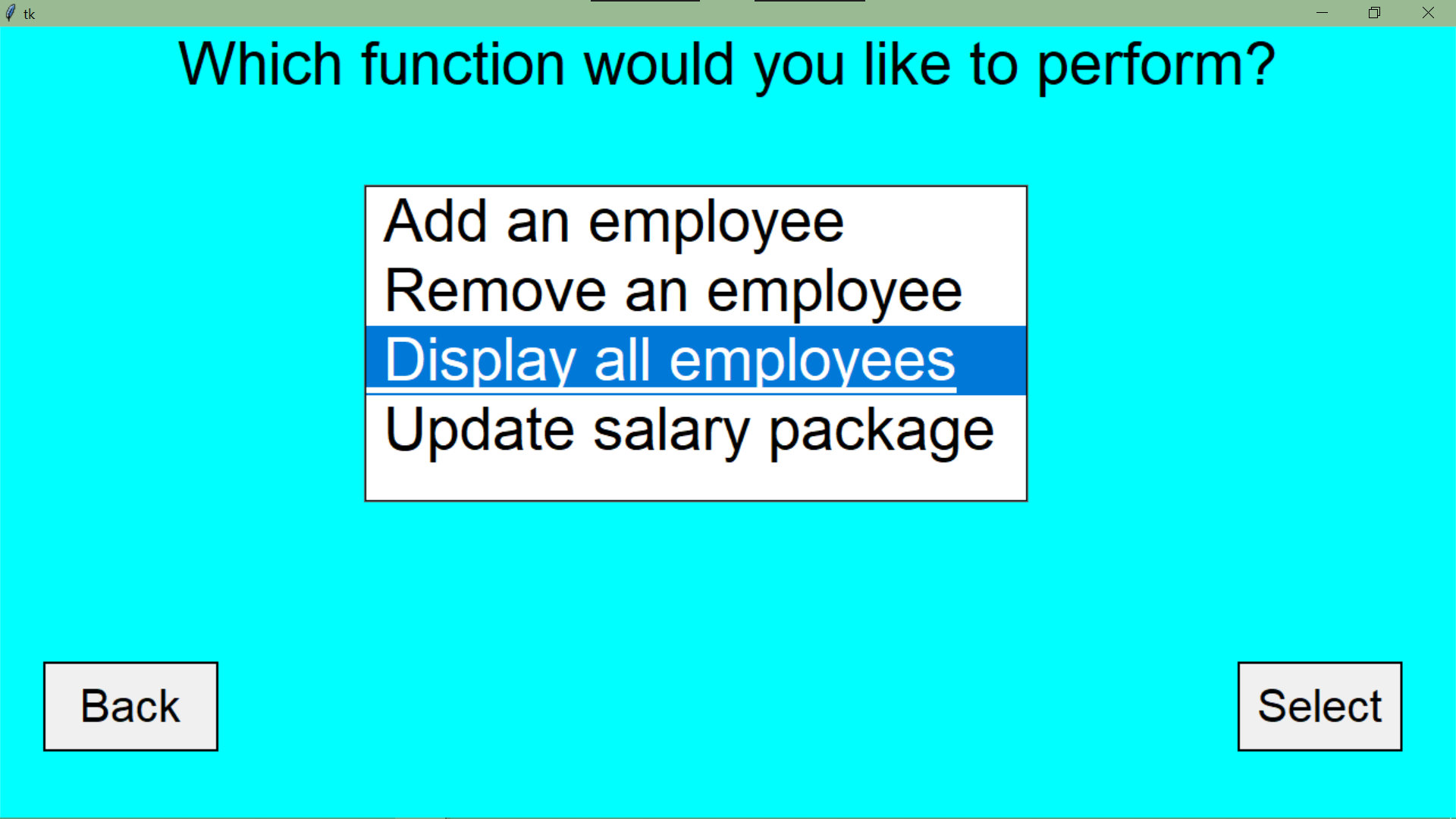


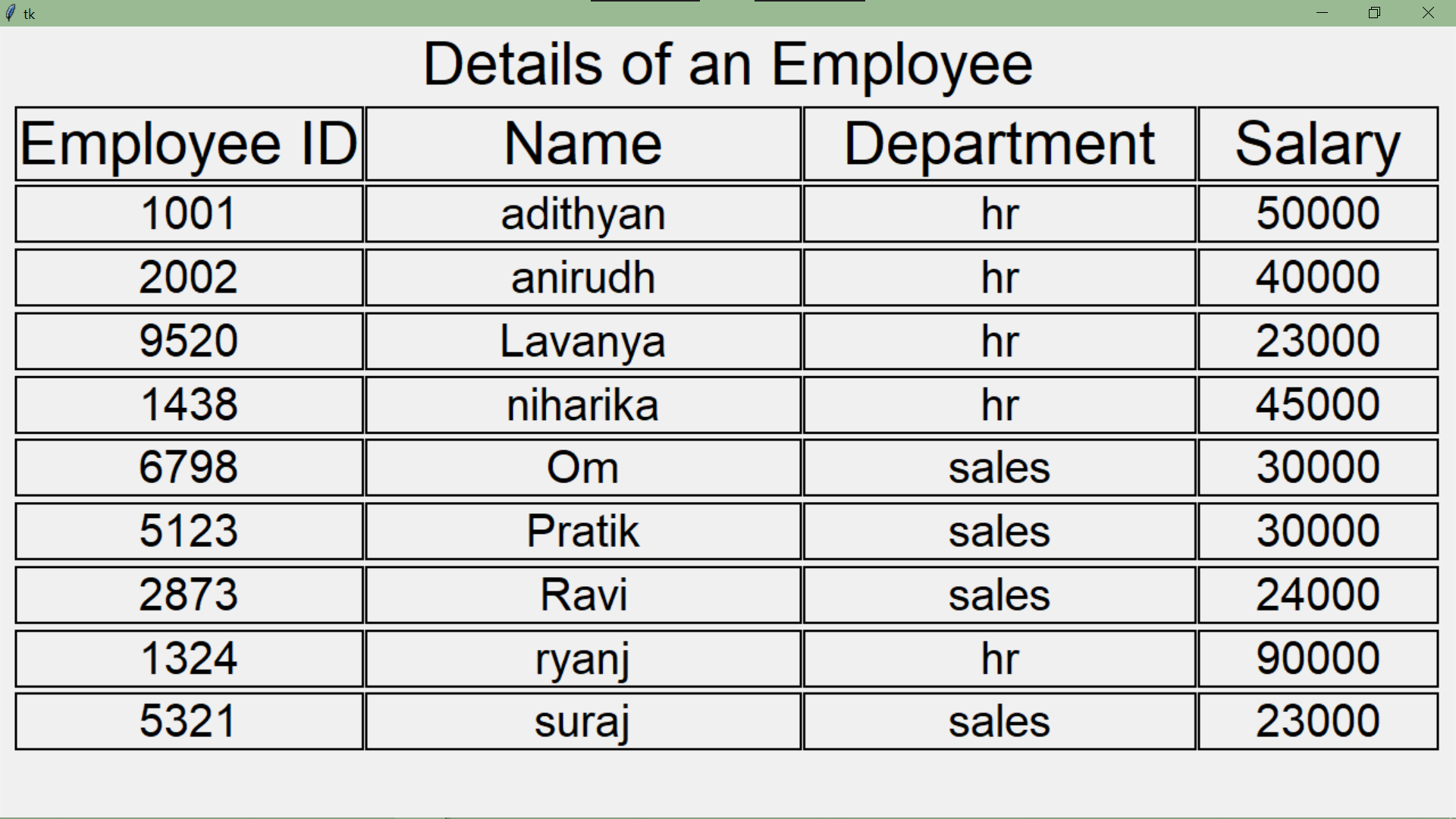




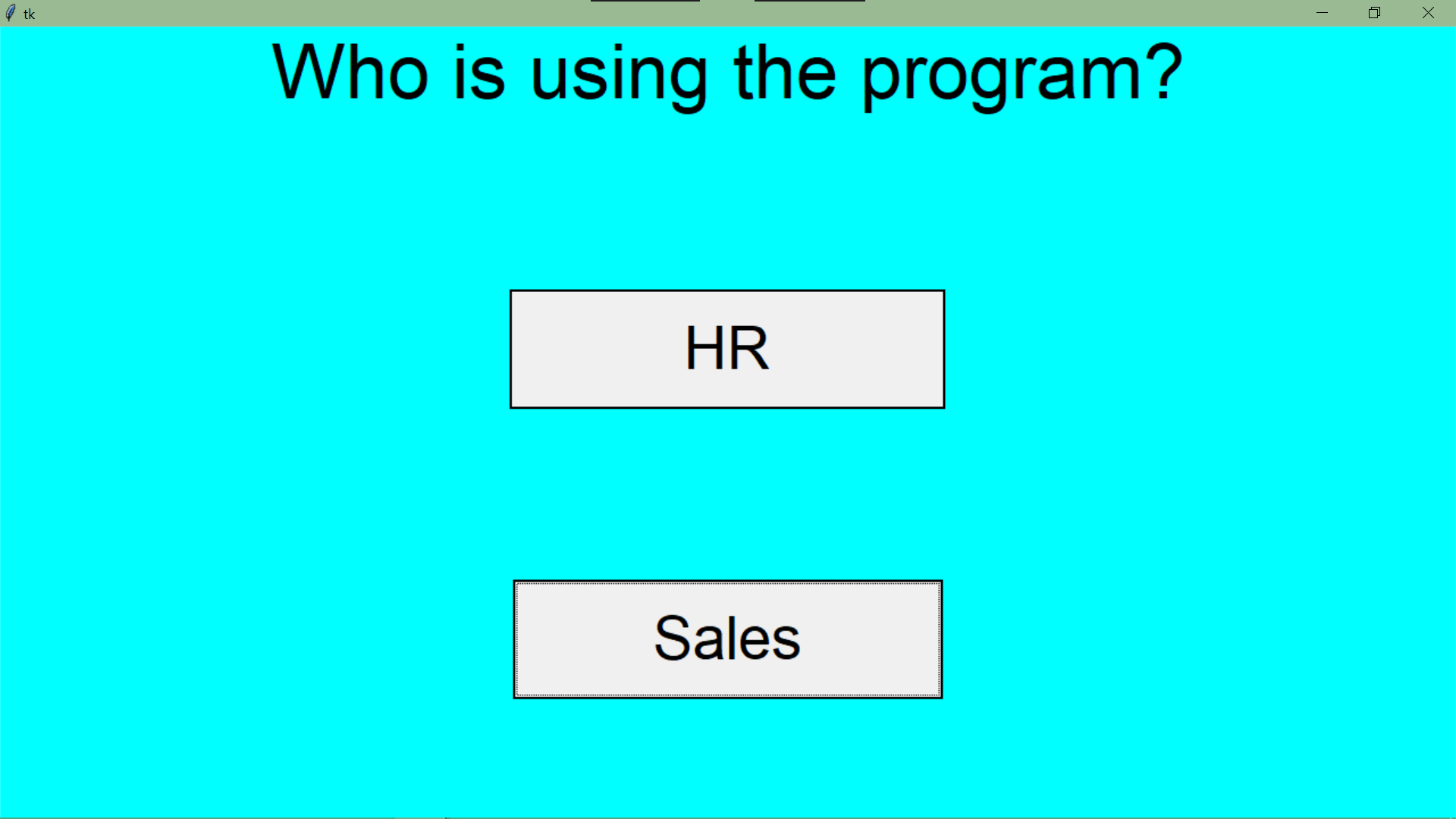












**#Sales.py**

from tkinter import \*

from tkinter import messagebox,ttk

from tkcalendar import Calendar

import mysql.connector as c

con=c.connect(host='localhost',user='root',passwd='achuadivava@1438',database='airinfo')

x=con.cursor()

SALES=Tk()

def back():

SALES.destroy()

import sec\_pg

SALES.configure(bg='cyan')

SALES.state('zoomed')

user\_label=Label(SALES,text='Username',bg='cyan',font=('Times Roman','20'))

user\_label.pack(pady=30)

user=Entry(SALES,font=('Papyrus','20'))

user.pack(padx='50')

pass\_label=Label(SALES,text='Password',bg='cyan',font=('Times Roman','20'))

pass\_label.pack(pady=30)

passw=Entry(SALES,font=('Papyrus','20'),show='\*')

passw.pack()

def check():

n=user.get()

p=passw.get()

query='select \* from salespass where usern="{}" and passw ="{}"'.format(n,p)

x.execute(query)

y=x.fetchone()

if n=='' or y==None:

messagebox.showwarning('Error','Invalid Username')

elif p=='':

messagebox.showerror('Error','Invalid Password')

else:

def no():

import sales\_2

def yes():

def nxt():

t=trans\_no.get()

f=flt.get()

i=inc.get()

def submit():

d=date.get\_date()

query="select transacnr from sales where transacnr='{}'".format(t)

x.execute(query)

e=x.fetchone()

if e!=None or t=='':

messagebox.showerror('Error','Transaction no not entered')

elif f=='':

messagebox.showerror('Error','Flight Name not entered')

elif i=='':

messagebox.showerror('Error','Amount received after transaction not entered')

elif d=='':

messagebox.showerror('Error','Date not entered')

else:

query1="insert into sales values('{}','{}','{}',{})".format(t,d,f,i)

x.execute(query1)

con.commit()

sales.destroy()

SALES.state('zoomed')

import sales\_2

transno\_label.destroy(),trans\_no.destroy(),inc\_label.destroy()

inc.destroy(),flt\_label.destroy(),flt.destroy()

date\_label=Label(sales,text='Date',bg='cyan',font=('',40))

date\_label.pack()

date=Calendar(sales,selectmode='day',date\_pattern='y-mm-dd',font=('',30))

date.pack()

btn.config(text='Submit',command=submit)

SALES.state('withdraw')

sales=Tk()

sales.configure(bg='cyan')

sales.state('zoomed')

transno\_label=Label(sales,text='Transaction No',bg='cyan',font=('',40))

transno\_label.place(relx=0.025,rely=0.02)

trans\_no=Entry(sales,font=('',30))

trans\_no.pack(anchor='e',padx=10,pady=30)

inc\_label=Label(sales,text='Amount received after transaction',bg='cyan',font=('',40))

inc\_label.place(relx=0.025,rely=0.18)

inc=Entry(sales,font=('',30))

inc.pack(anchor='e',padx=10,pady=30)

flt\_label=Label(sales,text='Name of Flight',bg='cyan',font=('',40))

flt\_label.place(relx=0.025,rely=0.36)

l=['AirIndia','Emirates','FlyDubai']

txt=StringVar()

flt=ttk.Combobox(sales,textvariable=txt,font=('',30))

flt['values']=l

flt.current(0)

flt.pack(anchor='e',padx=10,pady=30)

btn=Button(sales,text='Continue',font=('',40),command=nxt,relief='solid')

btn.place(rely=0.85,relx=0.4)

r1.destroy(),r2.destroy(),user\_label.destroy(),user.destroy()

pass\_label.destroy(),passw.destroy()

l1=Label(SALES,text='Welcome to the\nSales department of Airlink Infotech',bg='cyan',font=('',45))

l1.pack()

l2=Label(SALES,text='Are you using this program for the first time?',bg='cyan',font=('',40))

l2.pack(pady=200)

b1=Radiobutton(SALES,text='Yes',font=('',35),command=yes,relief='solid')

b1.pack(side='right')

b2=Radiobutton(SALES,text='No',font=('',35),command=no,relief='solid')

b2.pack(side='left',ipadx=10)

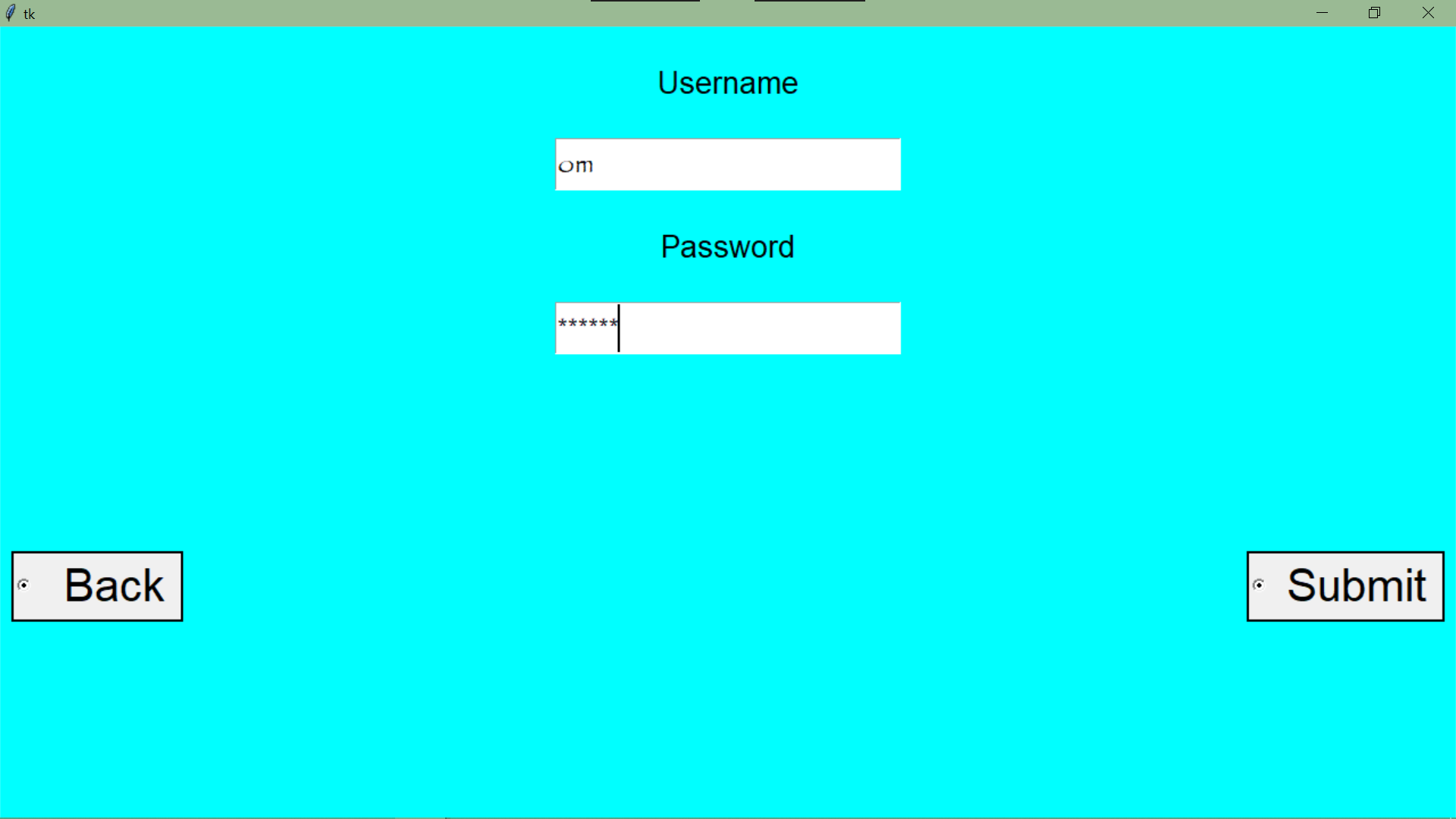
r1=Radiobutton(SALES,text='Submit ',font=('',30),command=check,relief='solid')

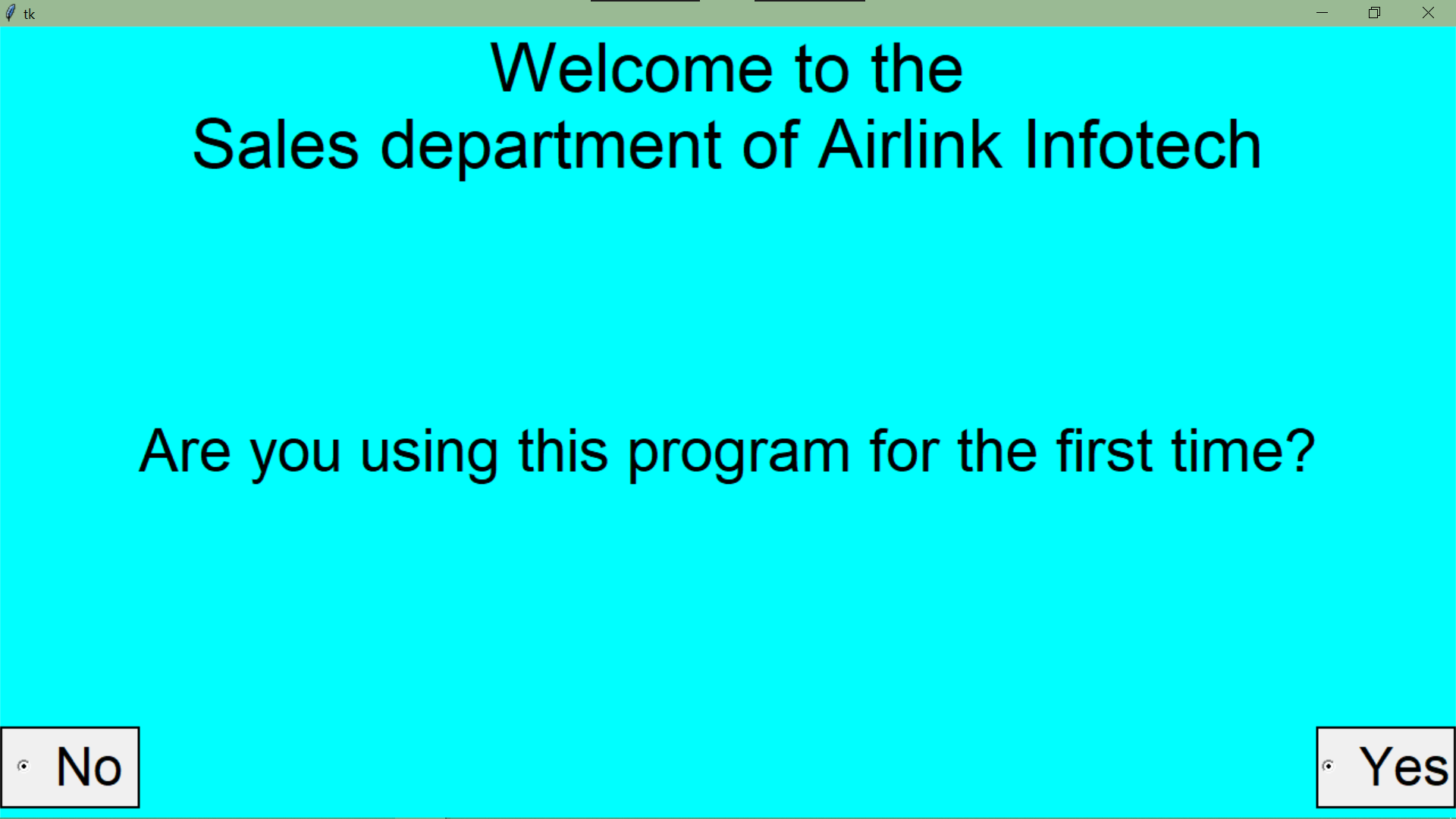
r1.pack(side='right',pady=30,padx=10)

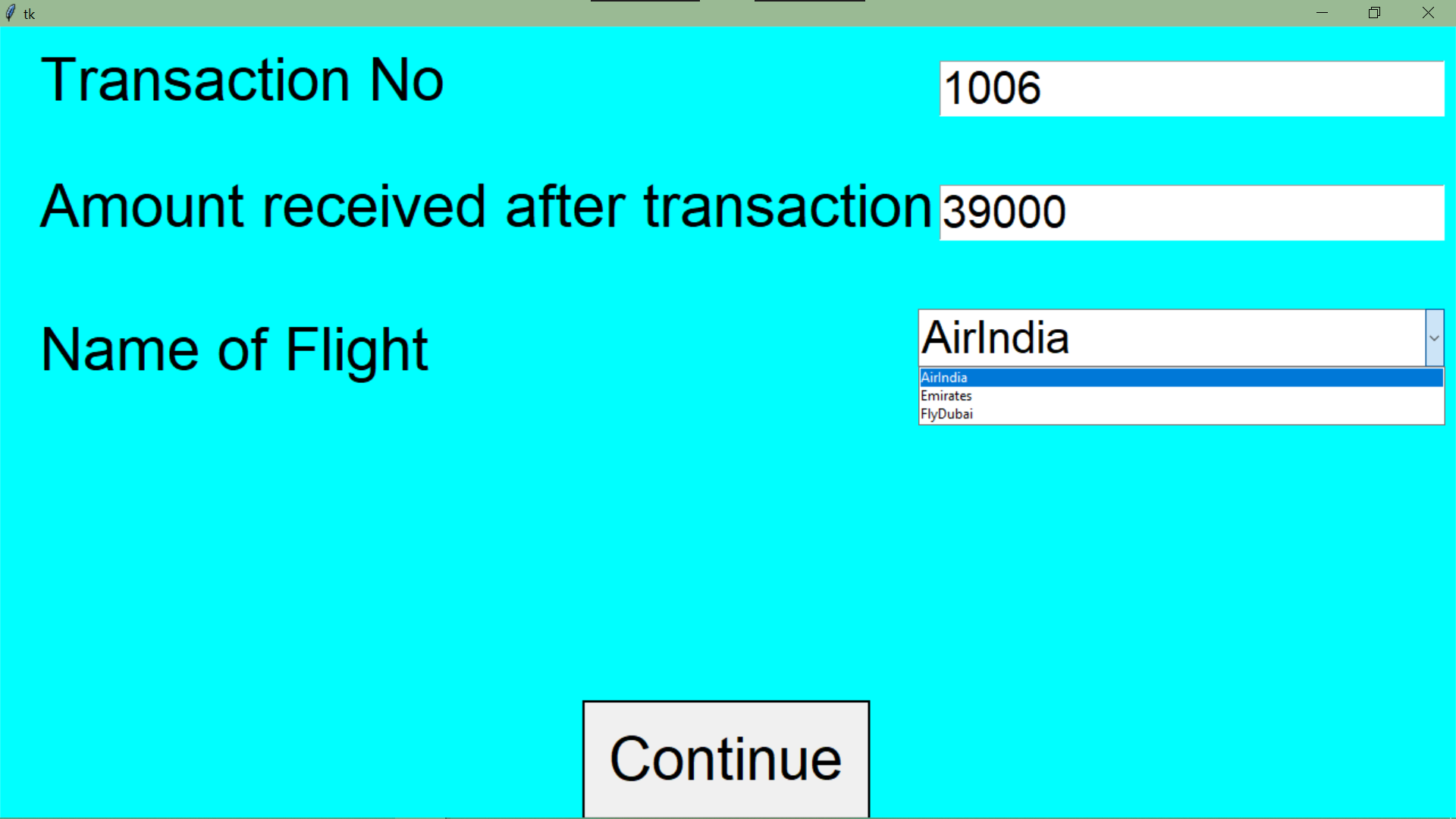
r2=Radiobutton(SALES,text=' Back ',font=('',30),command=back,relief='solid')

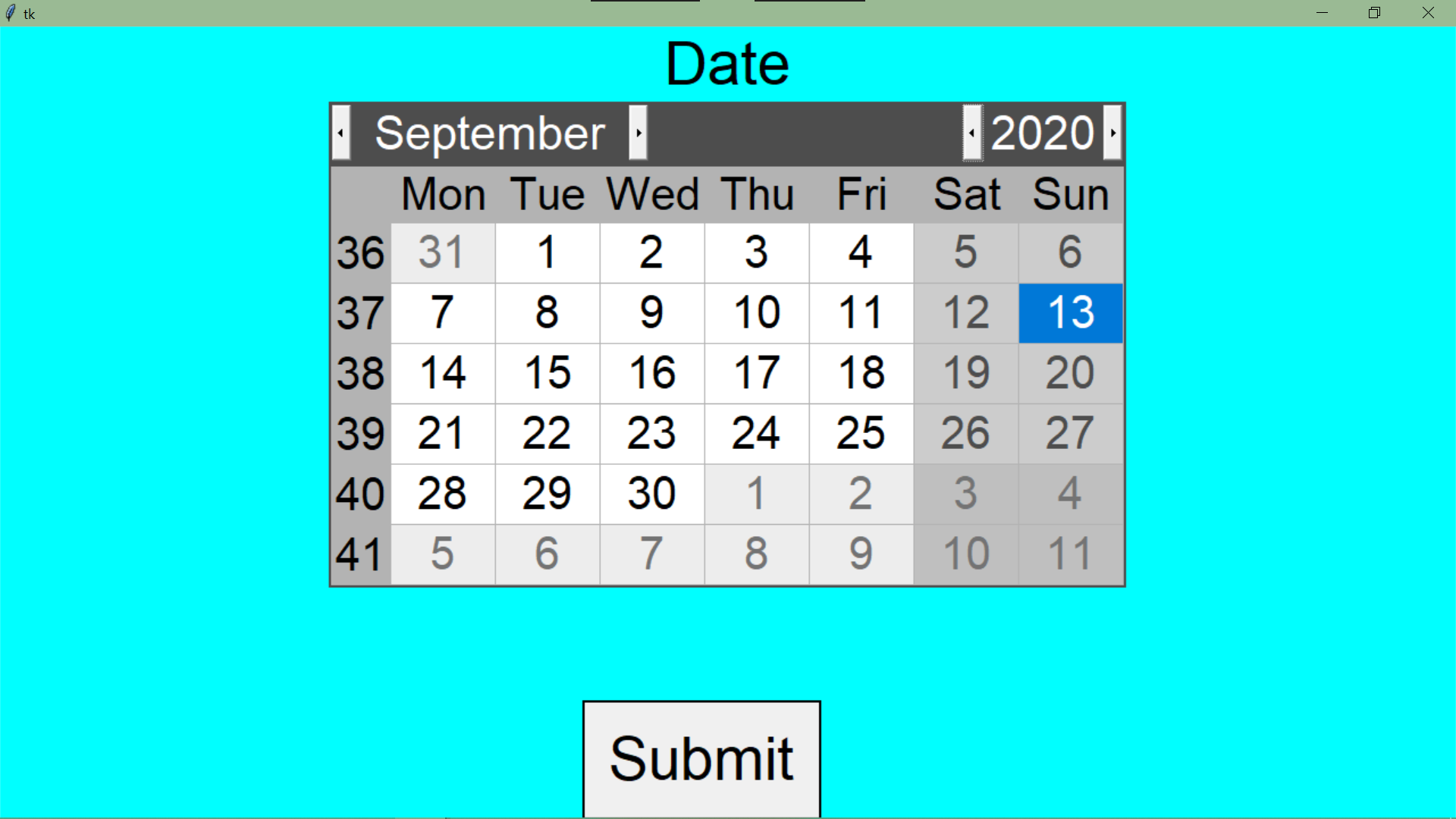
r2.pack(side='left',pady=30,padx=10)

SALES.mainloop()









**#sales\_2.py**

#Review the sales of a particular month:

#Of a flightfrom tkinter import \*

from tkinter import \*

from tkinter import ttk,messagebox

import graphit

import mysql.connector as c

con=c.connect(host='localhost',user='root',passwd='achuadivava@1438',database='airinfo')

x=con.cursor()

SALEs=Tk()

SALEs.config(bg='cyan')

SALEs.state('zoomed')

#Modify the sales of a particular transaction number

def \_1():

def submit():

i=inc.get()

t=transacnr.get()

j="select transacnr from sales where transacnr='{}'".format(t)

x.execute(j)

k=x.fetchone()

if i=='':

messagebox.showerror('Error','New Income not entered')

elif t=='' or len(k)==0:

messagebox.showerror('Error','Invalid Transaction No')

else:

query="update sales set SaleAmt={} where TransacNr='{}'".format(i,t)

x.execute(query)

con.commit()

messagebox.showinfo('Success','Modified Successfully')

modify.destroy()

modify=Tk()

modify.state('zoomed')

modify.config(bg='cyan')

l2=Label(modify,text='To Modify the sales of a particular transaction number',font=('',40),bg='cyan')

l2.pack(fill='x')

l1=Label(modify,bg='cyan',text='Transaction No',font=('Times Roman',40))

l1.place(relx=0.1,rely=0.2)

transacnr=Entry(modify,font=('',22))

transacnr.pack(anchor='ne',padx=100,pady=95)

l3=Label(modify,bg='cyan',text='New Income',font=('',40))

l3.place(relx=0.1,rely=0.4)

inc=Entry(modify,font=('',22))

inc.pack(anchor='ne',padx=100)

b=Button(modify,text='Submit',font=('',40),command=submit,relief='solid')

b.pack(side='bottom')

#Get the total sales of the year of

#i a flight

#ii airlink Infotech

def \_2():

sales=Tk()

sales.config(bg='cyan')

sales.state('zoomed')

def flight(): #Of flight

F=Tk()

F.config(bg='cyan')

F.state('zoomed')

def nxt():

f=flt.get()

ye=yer.get()

if f not in ('AirIndia','Emirates','FlyDubai'):

messagebox.showerror('Error','Employee no not entered')

elif ye not in ('2019','2020','2021'):

messagebox.showerror('Error','Employee Name not entered')

else:

l3.destroy(),l1.destroy(),l2.destroy(),b1.destroy(),yer.destroy(),flt.destroy()

query="select SaleAmt from sales where year(DateTransac)='{}' and FlightNm='{}'".format(ye,f)

x.execute(query)

y=x.fetchall()

sum1=0

for i in y:

sum1+=i[0]

L=Label(F,text='The total sales of '+f+' for the year '+ye+' is ₹'+str(sum1),font=('',40),bg='cyan')

L.pack(anchor='center')

l3=Label(F,text="Get the total sales of the year of a flight",font=('Times Roman',40),bg='cyan')

l3.pack(fill='x')

l1=Label(F,text='Name of the flight',bg='cyan',font=('',40))

l1.place(relx=0.1,rely=0.2)

l=['AirIndia','Emirates','FlyDubai']

txt=StringVar()

flt=ttk.Combobox(F,textvariable=txt,font=('',30))

flt['values']=l

flt.current(0)

flt.pack(anchor='ne',padx=100,pady=80)

l2=Label(F,text='Year to be viewed',bg='cyan',font=('',40))

l2.place(relx=0.1,rely=0.4)

yer=Entry(F,font=('',30))

yer.pack(anchor='ne',padx=100,pady=10)

b1=Button(F,text='Next',font=('',40),command=nxt,relief='solid')

b1.pack(side='bottom',pady=20)

def airlink(): #Of Airlink Infotech

airlink=Tk()

airlink.config(bg='cyan')

airlink.state('zoomed')

def nxt():

year=yer.get()

if year=='':

messagebox.showerror('Error','Year to be viewed not entered')

else:

l1.destroy(),b1.destroy(),yer.destroy()

query="select SaleAmt from sales where year(DateTransac)='{}'".format(year)

x.execute(query)

y=x.fetchall()

sum1=0

for i in y:

sum1+=i[0]

Label(airlink,text="The total sales of Airlink Infotech\nfor the year "+year+" is ₹"+str(sum1),bg='cyan',font=('',50)).pack(anchor=NSEW)

l=Label(airlink,text="Get the total sales of the year of Airlink Infotech",font=('',40),bg='cyan')

l.pack()

l1=Label(airlink,text='Year to be viewed',bg='cyan',font=('',40))

l1.place(relx=0.1,rely=0.5)

yer=Entry(airlink,font=('',30))

yer.pack(padx=90,side='right')

b1=Button(airlink,text='Next',font=('',40),command=nxt,relief='solid')

b1.pack(side='bottom')

l1=Label(sales,text="Get the total sales of the year",bg='cyan',font=('',40))

l1.pack()

b1=Button(sales,text='Of a flight',font=('',40),command=flight,relief='solid')

b1.pack(pady=100,ipadx=90)

b2=Button(sales,text='Of Airlink Infotech',font=('',40),command=airlink,relief='solid')

b2.pack()

#Compare year's sales of different flights through graphs

def \_3():

YR=Tk()

YR.config(bg='cyan')

YR.state('zoomed')

def nxt():

yr=yer.get()

if yr=='':

messagebox.showerror('Error','Year to be viewed not entered')

else:

query="select FlightNm, month(DateTransac),SaleAmt from sales where year(DateTransac)='{}'".format(yr)

x.execute(query)

y=x.fetchall()

dict1={}

for i in y:

dict1[(i[0],i[1])]=i[2]

A={}

E={}

F={}

month={1:'Jan',2:'Feb',3:'Mar',4:'Apr',5:'May',6:'Jun',7:'Jul',8:'Aug',9:'Sep',10:'Oct',11:'Nov',12:'Dec'}

for i in month.values():

A.setdefault(i,0)

E.setdefault(i,0)

F.setdefault(i,0)

for i,j in dict1:

if i=="AirIndia":

A[month[j]]+=dict1[(i,j)]

elif i=='Emirates':

E[month[j]]+=dict1[(i,j)]

else:

F[month[j]]+=dict1[(i,j)]

AirIndia=list(A.values())

Emirates=list(E.values())

FlyDubai=list(F.values())

graphit.bar(AirIndia,Emirates,FlyDubai)

l1=Label(YR,text="Compare this year's sales of different\nflights through a graphs",bg='cyan',font=('',40))

l1.pack()

l2=Label(YR,text='Year to be viewed',font=('Times Roman',40),bg='cyan')

l2.pack(pady=100)

yer=Entry(YR,font=('',22))

yer.pack()

b1=Button(YR,text='Submit',font=('',40),command=nxt,relief='solid')

b1.pack(side='bottom',pady=20)

#Dispaly the sales database

def \_4():

db=Tk()

db.state('zoomed')

query="select \* from sales"

x.execute(query)

y=x.fetchall()

L=Label(db,text='Sales Database',font=('',40))

L.place(relwidth=1)

L1=Label(db,text='Transaction No',font=('',30),borderwidth=2,relief='ridge')

L1.place(rely=0.1,relx=0.01)

L2=Label(db,text='Date of Transaction',font=('',30),borderwidth=2,relief='ridge')

L2.place(rely=0.1,relx=0.225,relwidth=0.3)

L3=Label(db,text='Flight Name',font=('',30),borderwidth=2,relief='ridge')

L3.place(rely=0.1,relx=0.527,relwidth=0.25)

L4=Label(db,text='Sales Amount',font=('',30),borderwidth=2,relief='ridge')

L4.place(rely=0.1,relx=0.78,relwidth=0.21)

j=0.18

for i in y:

l1=Label(db,text=i[0],font=('',20),borderwidth=2,relief='ridge')

l1.place(rely=j,relx=0.01,relwidth=0.21)

l2=Label(db,text=i[1],font=('',20),borderwidth=2,relief='ridge')

l2.place(rely=j,relx=0.225,relwidth=0.3)

l3=Label(db,text=i[2],font=('',20),borderwidth=2,relief='ridge')

l3.place(rely=j,relx=0.527,relwidth=0.25)

l4=Label(db,text=i[3],font=('',20),borderwidth=2,relief='ridge')

l4.place(rely=j,relx=0.78,relwidth=0.21)

j+=0.06

#Review the sales of a particular month

def \_5():

m=Tk()

def show():

lst=['AirIndia','Emirates','Fly Dubai']

a=l.index(month.get())

b=yr.get()

c=l1[box.curselection()[0]]

query="select SaleAmt from sales where FlightNm='{}' and year(DateTransac)='{}' and month(DateTransac)='{}'".format(c,b,a+1)

x.execute(query)

y=x.fetchall()

sum1=0

new=Tk()

new.state('zoomed')

new.config(bg='cyan')

for i in y:

sum1+=i[0]

Label(new,text="The sale made by "+str(c)+" in "+str(l[a])+" of\n"+str(b)+" is ₹"+str(sum1),font=('',40),bg='cyan').pack()

m.state('zoomed')

m.config(bg='cyan')

mon=Label(m,text='Select the month',bg='cyan',font=('',30))

mon.place(relx=0.2)

l=['January','February','March','April','May','June','July','August','September','October','November','December']

txt=StringVar()

month=ttk.Combobox(m,textvariable=txt,font=('',30))

month['values']=l

month.current(0)

month.pack(padx=200,pady=10,anchor='ne')

yr\_label=Label(m,text='Select year',bg='cyan',font=('',30))

yr\_label.place(relx=0.2,rely=0.15)

yr=Spinbox(m,from\_=2019,to=2021,textvariable='enter year',font=('',30))

yr.pack(padx=200,pady=20,anchor='ne')

box\_label=Label(m,text='Flight Name',bg='cyan',font=('',30))

box\_label.place(relx=0.2,rely=0.45)

box=Listbox(m,font=('',20))

box.pack(pady=30,anchor='ne',padx=200)

l1=['AirIndia','Emirates','FlyDubai']

box.insert(0,l1[0])

box.insert(1,l1[1])

box.insert(2,l1[2])

cont=Button(m,text='Continue',font=('',40),command=show,relief='solid')

cont.pack(side='bottom')

#Display the flight with the highest sales in a

# i.Month ii.Year

def \_6():

sales=Tk()

sales.config(bg='cyan')

sales.state('zoomed')

def flight(): #Month

F=Tk()

F.config(bg='cyan')

F.state('zoomed')

def nxt():

f=l.index(month.get())

ye=yr.get()

L.destroy(),mon.destroy(),month.destroy(),yr\_label.destroy(),yr.destroy(),b1.destroy()

query="select \* from sales where saleamt=(select max(saleamt) from sales where year(datetransac)={} and month(datetransac)='{}')".format(ye,f+1)

x.execute(query)

y=x.fetchone()

n=Label(F,text=y[2]+' has made the highest sale in '+l[f]+'\nof '+ye+' with a total sale of ₹'+str(y[3]),font=('',40),bg='cyan')

n.pack()

L=Label(F,text="Display the flight with the highest sales in a month",font=('',40),bg='cyan')

L.pack(fill='x')

mon=Label(F,text='Select the month',bg='cyan',font=('',30))

mon.place(relx=0.2,rely=0.1)

l=['January','February','March','April','May','June','July','August','September','October','November','December']

txt=StringVar()

month=ttk.Combobox(F,textvariable=txt,font=('',30))

month['values']=l

month.current(0)

month.pack(padx=200,pady=10,anchor='ne')

yr\_label=Label(F,text='Select year',bg='cyan',font=('',30))

yr\_label.place(relx=0.2,rely=0.25)

yr=Spinbox(F,from\_=2019,to=2021,textvariable='enter year',font=('Times Roman',30))

yr.pack(padx=200,pady=20,anchor='ne')

b1=Button(F,text='Next',font=('',40),command=nxt,relief='solid')

b1.pack(side='bottom')

def airlink(): #Year

F=Tk()

F.config(bg='cyan')

F.state('zoomed')

def nxt():

ye=yr.get()

L.destroy(),yr\_label.destroy(),yr.destroy(),b1.destroy(),b2.destroy()

query="select \* from sales where saleamt=(select max(saleamt) from sales where year(datetransac)='{}')".format(ye)

x.execute(query)

y=x.fetchone()

n=Label(F,text=y[2]+' has made the highest sale in\n'+ye+' with a total sale of ₹'+str(y[3]),font=('',40),bg='cyan')

n.pack()

L=Label(F,text="Display the flight with the highest sales in an year",bg='cyan',font=('Times Roman',40))

L.pack(fill='x')

yr\_label=Label(F,text='Select year',bg='cyan',font=('',30))

yr\_label.place(relx=0.2,rely=0.1)

yr=Spinbox(F,from\_=2019,to=2021,textvariable='enter year',font=('',30))

yr.pack(padx=200,pady=20,anchor='ne')

b1=Button(F,text='Next',font=('',40),command=nxt,relief='solid')

b1.pack(side='bottom')

l1=Label(sales,text="Display the flight with the highest sales in a",bg='cyan',font=('',40))

l1.pack()

b1=Button(sales,text='Month',font=('',40),command=flight,relief='solid')

b1.pack(pady=100)

b2=Button(sales,text='Year',font=('',40),command=airlink,relief='solid')

b2.pack(ipadx=15)

#Compare monthly sales of all flights through a graphical representaion

def \_7():

tot=Tk()

tot.state('zoomed')

tot.config(bg='cyan')

def nxt():

yer=yr.get()

mn=l.index(month.get())

TOT=Tk()

TOT.state('zoomed')

TOT.config(bg='cyan')

def pie():

graphit.piechart(list1,list2)

def bar():

graphit.bargraph(list1,list2)

query="select \* from sales where year(Datetransac)='{}' and month(Datetransac)='{}'".format(yer,mn+1)

x.execute(query)

y=x.fetchall()

list1=['AirIndia','Emirates','FlyDubai']

list2=[0,0,0]

for i in y:

if i[2]=="AirIndia":

list2[0]+=i[3]

elif i[2]=="Emirates":

list2[1]+=i[3]

else:

list2[2]+=i[3]

L1=Label(TOT,text='Compare monthly sales of all flights through a',bg='cyan',font=('',40))

L1.pack(fill='x')

B1=Button(TOT,text='Pie Graph',font=('',40),command=pie,relief='solid')

B1.pack(pady=50)

b2=Button(TOT,text='Bar Graph',font=('',40),command=bar,relief='solid')

b2.pack()

l1=Label(tot,text="Compare monthly sales of all flights through a graph",bg='cyan',font=('',40))

l1.pack()

mon=Label(tot,text='Select the month',bg='cyan',font=('',30))

mon.place(relx=0.2,rely=0.1)

l=['January','February','March','April','May','June','July','August','September','October','November','December']

txt=StringVar()

month=ttk.Combobox(tot,textvariable=txt,font=('',30))

month['values']=l

month.current(0)

month.pack(padx=200,pady=10,anchor='ne')

yr\_label=Label(tot,text='Select year',bg='cyan',font=('',30))

yr\_label.place(relx=0.2,rely=0.35)

yr=Spinbox(tot,from\_=2019,to=2021,textvariable='enter year',font=('',30))

yr.pack(padx=200,pady=100,anchor='ne')

b1=Button(tot,text='Next',font=('',40),command=nxt,relief='solid')

b1.pack(side='bottom')

def home():

SALEs.destroy()

import sec\_pg

def nxt():

a=lb.curselection()

ind=lb.index(a)

if ind==0:

\_1()

elif ind==1:

\_2()

elif ind==2:

\_3()

elif ind==3:

\_4()

elif ind==4:

\_5()

elif ind==5:

\_6()

else:

\_7()

label=Label(SALEs,text='Which function would you like to perform?',bg='cyan',font=('',30))

label.pack()

lb=Listbox(SALEs,font=('',30),fg='black',relief='solid')

lb.pack(pady=5,fill='x',padx=5)

lb.insert(0,'Modify the sales of a particular transaction number')

lb.insert(1,'Get the total sales of the year')

lb.insert(2,"Compare the sales of different flights for a particular month through graphs")

lb.insert(3,'Dispaly the sales database')

lb.insert(4,'Review the sales of a particular month')

lb.insert(5,'Display the flight with the highest sales in a')

lb.insert(6,'Compare monthly sales of all flights through a graph')

slc=Button(SALEs,text='Select',font=('',20),command=nxt,relief='solid')

slc.place(relx=0.9,rely=0.9)

back=Button(SALEs,text='Back',font=('',20),command=home,relief='solid')

back.place(rely=0.9,relx=0.02,relwidth=0.08)

SALEs.mainloop()

**#graphit.py**

import matplotlib.pyplot as plt

import numpy as np

def bar(AirIndia,Emirates,FlyDubai):

w=0.2

x=['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec']

bar1=np.arange(len(x))

bar2=[i+w for i in bar1]

bar3=[i+w for i in bar2]

plt.bar(bar1,AirIndia,w,label="AirIndia")

plt.bar(bar2,Emirates,w,label="Emirates")

plt.bar(bar3,FlyDubai,w,label="FlyDubai")

plt.xlabel("Months")

plt.ylabel("Revenue")

plt.title("Sales Report")

plt.xticks(bar1+w,x)

plt.legend()

plt.show()

def piechart(list1,list2):

plt.style.use("fivethirtyeight")

a=max(list2)

explode=[0,0,0]

b=list2.index(a)

explode[b]=0.3

plt.pie(list2, labels=list1, explode=explode, shadow=True,startangle=90,autopct='%1.1f%%',wedgeprops={'edgecolor':'black'})

plt.title("Monthly sales of the Flights")

plt.tight\_layout()

plt.legend()

plt.show()

def linegraph(yer,AirIndia,Emirates,FlyDubai):

plt.title("Sales of three flights for the year",yer)

x=['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec']

plt.plot(x,AirIndia)

plt.plot(x,Emirates)

plt.plot(x,FlyDubai)

plt.legen()

plt.show()

def bargraph(list1,list2):

ypos=np.arange(len(list1))

plt.xticks(ypos,list1)

plt.xlabel("Flight Names")

plt.ylabel("Revenue Generated")

plt.bar(ypos,list2,label="Revenue")

plt.legend()

plt.show()

