```
#BASIC PENSION AND COMMUTATION CALCULATOR FOR BANK EMPLOYEES


#Imported tkinter for GUI

#Imported messagebox to display message in messagebox

#Imported tkinter as tk(OBJECT)

from tkinter import *

from tkinter import messagebox

import tkinter as tk


#Function to calculate the Pension and commutation

def calculate():


    #Used a TopLevel interface to get the input details

    calculationscreen = Toplevel()

    calculationscreen.title("Pension and Commutaion Calculator")

    calculationscreen.geometry("800x750")

    calculationscreen.config(bg="#FDD7E4")


    #Providing Heading to the Interface

    Heading = StringVar()

    Headinglabel = Label(calculationscreen, textvariable = Heading
,bg="#7E587E",width="1000",pady="10")

    Heading.set("Pension and Commutation Calculator for Bank Employees")

    Headinglabel.pack()


    #This String is initially left BLANK to seperate the Heading and other Details for input

    mainline = StringVar()

    mainlinelable =
Message(calculationscreen,textvariable=mainline,width="1000",fg="#800517",pady="20")

    mainline.set("")

    mainlinelable.pack()
    #Asking User Date of Birth
    enterdateofbirth = StringVar()
    enterdateofbirthlabel = Message(calculationscreen,textvariable =
enterdateofbirth,bg="#C9BE62",width="1000",pady="10")
    enterdateofbirth.set("Enter Date of Birth [DD/MM/YYYY] ")
    enterdateofbirthlabel.pack()
    dateofbirth = Entry(calculationscreen,bd=5,width="25")
    dateofbirth.pack()

    #Asking User Date of Retirement
    enterdateofretirement = StringVar()
    enterdateofretirementlabel = Message(calculationscreen,textvariable =
enterdateofretirement,bg="#C9BE62",width="1000",pady="10")
    enterdateofretirement.set("Enter Date of Retirement [DD/MM/YYYY] ")
    enterdateofretirementlabel.pack()
    dateofretirement = Entry(calculationscreen,bd=5,width="25")
    dateofretirement.pack()

    #Taking input of Qualifying Years i.e. No. of Years of Working
    enterqualifyingservice = StringVar()
    enterqualifyingservicelabel = Message(calculationscreen,textvariable =
enterqualifyingservice,bg="#C9BE62",width="1000",pady="10")
    enterqualifyingservice.set("Total Qualifying Service [Years] ")
    enterqualifyingservicelabel.pack()
    qualifyingservice = Entry(calculationscreen,bd=5,width="25")
    qualifyingservice.pack()

    #Taking Last Month's Salary as input
    entermonthsemoluments = StringVar()
    entermonthsemolumentslabel = Message(calculationscreen,textvariable =
entermonthsemoluments,bg="#C9BE62",width="1000",pady="20")
    entermonthsemoluments.set("Last Month's Emoluments [In Rs.] ")
    entermonthsemolumentslabel.pack()

    monthsemoluments = Entry(calculationscreen,bd=5,width="25")

    monthsemoluments.pack()


    #Taking Percent of Pension to be commuted

    entercommutedpension = StringVar()

    entermonthcommutedpensionlabel = Message(calculationscreen,textvariable =
entercommutedpension,bg="#C9BE62",width="1000",pady="20")

    entercommutedpension.set("Pension to be Commuted (in %) [Maximum 40%] ")

    entermonthcommutedpensionlabel.pack()

    commutedpension = Entry(calculationscreen,bd=5,width="25")

    commutedpension.pack()




    #This is the function for the calcualtion on the basis of input provided

    def pensionandcommutation():


        #Obtaining the data entered by the user

        qualifyingserviceget = float(qualifyingservice.get())

        monthsemolumentsget = float(monthsemoluments.get())

        commutedpensionget = float(commutedpension.get())

        commutedpensionget = commutedpensionget/100


        #Storing the Date of Requirement

        dor=""

        for i in range(6,10,1):

            dor=dor+dateofretirement.get()[i]


        #Storing the Date of Birth

        dob=""

        for i in range(6,10,1):
```

```
    dob=dob+dateofbirth.get()[i]

#Calculating the age at the retirement
age = int(int(dor)-int(dob))

#Finding the Pension and Commutation on the basis of "Qualifying Service"

if(qualifyingserviceget<10):
    basicpension=0
    enhanchedfamilypension=0
    normalfamilypension=0
    pensioncommuted=0
    reducedmonthlypensionaftercommutation=0
    totalcommutation=0

elif(qualifyingserviceget>=10 and qualifyingserviceget<25):
    basicpension=monthsemolumentsget/100
    enhanchedfamilypension=monthsemolumentsget/100*qualifyingserviceget
    normalfamilypension=monthsemolumentsget*0.3
    pensioncommuted=basicpension*commutedpensionget
    reducedmonthlypensionaftercommutation=basicpension-pensioncommuted
    totalcommutation=pensioncommuted*commutedpensionget*12

elif(qualifyingserviceget>=25):
    basicpension=monthsemolumentsget/50
    enhanchedfamilypension=monthsemolumentsget/2
    normalfamilypension=monthsemolumentsget*0.3
    pensioncommuted=basicpension*commutedpensionget
    reducedmonthlypensionaftercommutation=basicpension-pensioncommuted
    totalcommutation=pensioncommuted*commutedpensionget*12
```

```
        #Calculating the Additional Basic Pension


        if(age>=80 and age<85):
            additionalbasicpenion=basicpension*0.2
        elif(age>=85 and age<90):
            additionalbasicpenion=basicpension*0.3
        elif(age>=90 and age<95):
            additionalbasicpenion=basicpension*0.4
        elif(age>=95 and age<90):
            additionalbasicpenion=basicpension*0.5
        elif(age>=100):
            additionalbasicpenion=basicpension
        else:
            additionalbasicpenion=0


        #Displaying the Information in the messagebox
        messagebox.showinfo("Pension and Commutation","Basic Pension: " + str(basicpension) +
"\nEnhanced Family Pension: " + str(enhanchedfamilypension)
                            + "\nNormal Family Pension: " + str(normalfamilypension) + "\nAdditional Basic
Pension: " + str(additionalbasicpenion)
                            + "\nCommuted Pension: " + str(pensioncommuted) + "\nMonthly Reduced Pension
After Commutation: " + str(reducedmonthlypensionaftercommutation)
                            + "\nTotal Commutation: " + str(totalcommutation))



    #Adding the Details of the Employee in the file
    def addtolist():


        #Obtaining the data entered by the user
        qualifyingserviceget = float(qualifyingservice.get())
        monthsemolumentsget = float(monthsemoluments.get())
        commutedpensionget = float(commutedpension.get())
commutedpensionget = commutedpensionget/100

#Storing the Date of Requirement
dor=""
for i in range(6,10,1):
   dor=dor+dateofretirement.get()[i]

#Storing the Date of Birth
dob=""
for i in range(6,10,1):
   dob=dob+dateofbirth.get()[i]

#Calculating the age at the retirement
age = int(int(dor)-int(dob))

#Finding the Pension and Commutation on the basis of "Qualifying Service"

if(qualifyingserviceget<10):
   basicpension=0
   enhanchedfamilypension=0
   normalfamilypension=0
   pensioncommuted=0
   reducedmonthlypensionaftercommutation=0
   totalcommutation=0

elif(qualifyingserviceget>=10 and qualifyingserviceget<25):
   basicpension=monthsemolumentsget/100
   enhanchedfamilypension=monthsemolumentsget/100*qualifyingserviceget
   normalfamilypension=monthsemolumentsget*0.3
   pensioncommuted=basicpension*commutedpensionget
   reducedmonthlypensionaftercommutation=basicpension-pensioncommuted
```

```
        totalcommutation=pensioncommuted*commutedpensionget*12


      elif(qualifyingserviceget>=25):
        basicpension=monthsemolumentsget/50
        enhanchedfamilypension=monthsemolumentsget/2
        normalfamilypension=monthsemolumentsget*0.3
        pensioncommuted=basicpension*commutedpensionget
        reducedmonthlypensionaftercommutation=basicpension-pensioncommuted
        totalcommutation=pensioncommuted*commutedpensionget*12


      #Opening the File to add the Data
      f=open("EmployeesList.txt","a")
      f.write("  " + str(username) + "\t\t\t  "  + str(basicpension) +  "\t\t\t  " + 
str(enhanchedfamilypension) + "\t\t\t  "  + str(normalfamilypension) + "\t\t\t  " + 
str(pensioncommuted) + "\t\t\t  " + str(totalcommutation) + "   ")
      f.write("\n\n")
      f.close()
      messagebox.showinfo("Add to List","Information has been successfully added.")


    #This String is initially left BLANK to seperate the Heading and other Details for input
    mainline1 = StringVar()
    mainlinelable1 = 
Message(calculationscreen,textvariable=mainline1,width="1000",fg="#800517",pady="20")
    mainline1.set("")
    mainlinelable1.pack()


    #Button for Finding the Pension and Commutation
    estimatedpricebutton = Button(calculationscreen,text = "Click to See Pension and 
Commutation",bg="#2B65EC",fg="#82CAFA",padx="20",pady="20",command=pensionandcommutat
ion)
    estimatedpricebutton.pack()


    #This String is initially left BLANK to seperate the Heading and other Details for input
    mainline2 = StringVar()
    mainlinelable2 =
Message(calculationscreen,textvariable=mainline2,width="1000",fg="#800517",pady="20")
    mainline2.set("")
    mainlinelable2.pack()

    #Button for Adding the Details to the File
    addtolistbutton = Button(calculationscreen,text="Click to Add
Details",bg="#2B65EC",fg="#82CAFA",padx="20",pady="20",command=addtolist)
    addtolistbutton.pack()

#Function to show the List of Employees
def listofemployees():

    #Creating the TopLevel to Show the List of Employees to the screen
    listofemployees=Toplevel()
    listofemployees.geometry("1000x500")
    listofemployees.title("List of Employees")

    #Heading for this interface
    Heading = StringVar()
    Headinglabel = Label( listofemployees, textvariable = Heading
,bg="#7E587E",width="1000",pady="10")
    Heading.set("List of Employees")
    Headinglabel.pack()

    #Sub-Heading
    Headingnext = StringVar()
    Headingnextlabel = Label( listofemployees, textvariable = Headingnext
,bg="#46C7C7",width="1000",pady="10")
    Headingnext.set("  Employee Name    \t    Basic Pension     \t    Enhanced Family Pension     \t
Normal Family Pension    \t    Commuted Pension   \t   Total Commutation    ")
    Headingnextlabel.pack()
    #Showing the list of the employees and their details from the file "EmployeesList.txt"

    employeedetailsinstringcopy = StringVar()

    employeedetailsinstring=""

    f = open("EmployeesList.txt","r")

    employeedetails = f.readlines()

    for employee in employeedetails:

        employeedetailsinstring = employeedetailsinstring + employee

    f.close()

    employeelabel = Message(listofemployees,textvariable = 
employeedetailsinstringcopy,width="1000",pady="20")

    employeedetailsinstringcopy.set(employeedetailsinstring)

    employeelabel.pack()

    listofemployees.mainloop()


#This is the mainscreen consisiting of different options

def mainscreen():


    screen  = tk.Tk()

    screen.title("Main Screen")

    screen.geometry("600x400")

    screen.config(bg="#C6AEC7")


    #Heading for the Main Scren

    Heading = StringVar()

    Headinglabel = Label( screen, textvariable = Heading ,bg="#FBBBB9",width="200",pady="20")

    Heading.set("Pension and Commutation Calculator for Bank Employees")

    Headinglabel.pack()


    mainline = StringVar()

    mainlinelable = Message(screen,textvariable=mainline,width="1000",fg="#800517",pady="20")
    mainline.set("Select Your Choice")
    mainlinelable.pack()

    #Button for finding the Pension and Commutation
    pensionandcommutationcalculation = Button(screen,text="Calculate Pension and
Commutation",bg="#7D1B7E",fg="#FCDFFF",padx="20",pady="10",command=calculate)
    pensionandcommutationcalculation.pack()

    #This String is initially left BLANK to seperate the Heading and other Details for input
    mainline1 = StringVar()
    mainlinelable1 = Message(screen,textvariable=mainline1,width="1000",fg="#800517",pady="20")
    mainline1.set("")
    mainlinelable1.pack()

    #Button to see the List of Employees and their details
    employeelist = Button(screen,text="List of
Employees",bg="#990012",fg="#F7E7CE",padx="10",pady="10",command=listofemployees)
    employeelist.pack()

    #This String is initially left BLANK to seperate the Heading and other Details for input
    mainline2 = StringVar()
    mainlinelable2 = Message(screen,textvariable=mainline2,width="1000",fg="#800517",pady="20")
    mainline2.set("")
    mainlinelable2.pack()

    #Function to exit the mainscreen
    def exitscreen():
        screen.destroy()

    #Button to exit from the mainscreen
    exit =
Button(screen,text="Exit",bg="#2B65EC",fg="#B6B6B4",padx="30",pady="10",command=exitscreen)
```

```
    exit.pack()

    screen.mainloop()

#Function to Check the Username and Password when LogIn
def check(username,password):

    username=username+"\n"
    password=password+"\n"

    #Opening the File of Username
    f=open("Username.txt","r")
    usernames = f.readlines()

    #"nameindex" to store the index at which name is found
    nameindex=0

    #"namefound" to confirm that the name is present
    namefound=0

    for name in usernames:
        nameindex=nameindex+1
        if(username==name):
            namefound=1
            break
    f.close()

    #Checking weather the name is present or not
    if namefound==0:
        return False
    #Opening the file of Password
    f=open("Password.txt","r")
    passwords = f.readlines()

    passwordindex=0

    for passs in passwords:
        passwordindex=passwordindex+1
        if(passwordindex==nameindex):
            if(passs==password):
                return True

    return False

#Asking the choice of the User
print("-------------Pension and Commutaion Calculator-----------")
print("Select your choice")
print("1.Login")
print("2.SignUp")
choice=input()

if(choice=='1'):
    print("Enter the username: ",end="")
    username=input()
    print("Enter the password: ",end="")
    password=input()

    #Validating the Username and Password
    if(check(username,password)==True):
        mainscreen()
    else:
        print("Input credentials are incorrect")



elif(choice=='2'):
    print("Enter the username: ",end="")
    username=input()
    print("Enter the password: ",end="")
    password=input()
    print("Re-enter the password: ",end="")
    repassword=input()

    #Checking weather password and re-entered password are Matching or Not
    if(password!=repassword):
        print("------Password and Re-entered Password are Not Matching------")

    #Storing the Username and Password in "Username.txt" and "Password.txt" respectively
    else:
        f=open("Username.txt","a")
        f.write(username)
        f.write("\n")
        f.close()
        f=open("Password.txt","a")
        f.write(password)
        f.write("\n")
        f.close()
        mainscreen()

#If selectef choice is incorrect
else:
    print("Input Choice is Incorrect")
```