



REPORT ON PYTHON PROGRAMMING

INT-213

PENSION AND COMMUTATION CALCULATOR | GROUP-10

PENSION AND COMMUTATION CALCULATOR

END-TERM REPORT

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

BY:

S.NO	Name	Roll No	Registration No
1.	ABHIJIT BISWAS	13	11905339
2.	KUMAWAT SHRIRAM BABULAL	03	11905309
3.	G. VISHNUPRIYA RAJAN	05	11905258

Course Code: INT213



School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)



OBJECTIVE

The primary objective of this project is to implement what we have learnt throughout our course of Python programming and use that to develop a Graphical User Interface (GUI) for Pension And Commutation Calculator with all the required functionalities.

This project also aims at providing a user-friendly interface to the users to let them easily use the Pension and Commutation Calculator, so that the lack of knowledge in term of Pension and Commutation can be destroyed.

Through this project we aim to target the elder/Aged part of the society and let them know that how much they lack in terms of the knowledge of their Pension and Commutation Scheme.

This project will also help in spreading awareness regarding the scheme introduced by the government for the pensioners.



ACKNOWLEDGEMENT

We want to thank our course teacher who believed in us and assigned us this project on python GUI "Pension And Commutation Calculator". We also want to thank other all who have helped us in making the project and completing it within the time frame

Abhijit Biswas
Babulal Shriram Kumawat.
G. Vishnupriya Rajan.



INTRODUCTION

PENSION:

A pension is a fund or can be said a beneficial fund where a fixed sum of amount is paid regularly to a person after their retirement. There are different types of pensions.

COMMUTATION:

If the person opts for commutation of pension then a portion of pension (max 40%) is cut off. The monthly sum is reduced by the pension commuted and will be restored on the expiry of 15 years from the date of receipt of the commuted value of pension.

CALCULATIONS:

For basic pension calculation:

Basic Pension = (One Month Emolument x Qualifying Service) x 70/1000

For commutation calculation:

Monthly Commutation = Commutation % x Basic pension

Total Commutation = (Commutation % x Basic pension) x 10years

Pension received by the person:

Reduced monthly pension = basic pension – Monthly Commutation





TERMINOLOGIES

Tkinter : Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

TTK : This module provides classes to allow using Tk themed widget set. Its basic idea is to separate, to the extent possible, the code implementing a widget's behavior from the code implementing its appearance.

PIL: PIL is the Python Imaging Library which provides the python interpreter with image editing capabilities. **PIL.Image.new()** method creates a new image with the given mode and size.

Messagebox: The tkMessageBox module is used to display message boxes in your applications. This module provides a number of functions that you can use to display an appropriate message. Some of these functions are showinfo, showwarning, showerror, askquestion, askokcancel, askyesno, and askretryignore.

Tkinter.font : You can create a "font object" by importing the tkFont module and using its Font class constructor. As a tuple whose first element is the font family, followed by a size in points, optionally followed by a string containing one or more of the style modifiers bold, italic, underline and overstrike.

Geometry : Tkinter provides many methods; one of them is the **geometry()** method. This method is used to set the dimensions of the [Tkinter](#) window and is used to set the position of the main window on the user's desktop.

Title: Title function in python is the Python String Method which is used to convert the first character in each word to Uppercase and remaining characters to Lowercase in string and returns new string.

Label : Tkinter Label is a widget that is used to implement display boxes where you can place text or images. The text displayed by this widget can be changed by the developer at any time you want.

Stringvar() : **StringVar()** is a class from tkinter. It's used so that you can easily monitor changes to tkinter variables if they occur through the example code provided: `def callback(*args): print "variable changed!" StringVar() # Holds a string; default value.`

IntVar() : These Tkinter control variables are used like regular **Python** variables to keep certain values. `intvar() # Holds an integer; default value 0.`

Entry: The Entry widget is used to accept single-line text String from a user.

- If you want to display multiple lines of text that can be edited, then you should use the *Text* widget.
- If you want to display one or more lines of text that cannot be modified by the user, then you should use the *Label* widget.

Combobox : Combobox is a combination of Listbox and an entry field. It is one of the Tkinter widgets where it contains a down arrow to select from a list of options. It helps the users to select according to the list of options displayed.

Textvariable : Associates a **Tkinter** variable (usually a StringVar) to the contents of the entry field. (**textVariable/Variable**) `validate=` Specifies when validation should be done.

Button : The Button widget is used to add buttons in a Python application. These buttons can display text or images that convey the purpose of the buttons. You can attach a function or a method to a button which is called automatically when you click the button.

Command : Execute the **Python** code in **command**. **command** can be one or more statements separated by newlines, with significant leadingwhitespace as in normal module code.

.place() : The **Place** geometry manager is the simplest of the three general geometry managers provided in Tkinter. It allows you explicitly set the position and size of a window, either in absolute terms, or relative to another window.

You can access the **place** manager through the **place()** method which is available for all standard widgets.



SOURCE CODE

1. Basic Pension Calculator:

```
# importing tkinter module.
import tkinter
from tkinter import ttk
from PIL import ImageTk, Image
from tkinter import messagebox #for pop-up window
import mysql.connector as sql #to connect to MySQL server
import tkinter.font as font

import os
top = tkinter.Tk()

# Give basic characteristics and geometry to the window.
top.geometry("1000x600")
image = Image.open('Basic_PensionImage.png')
photo_image = ImageTk.PhotoImage(image)
label = tkinter.Label(top, image = photo_image)
label.pack()
top.title(" - Pension and Commutation Calculator - ")

#declaring the variables required for storing and calculating the values.
name_var=tkinter.StringVar()
year_var=tkinter.IntVar()
year_var=tkinter.IntVar()
one_var=tkinter.IntVar()
ten_var=tkinter.IntVar()
commutation_var=tkinter.IntVar()
basic_pension= tkinter.IntVar()
reduced_pension= tkinter.IntVar()
totalcommutation_var= tkinter.IntVar()

# creating a label for title using widget Label
```



```

title_label = tkinter.Label(top, text = ' - Basic Pension Calculator - ',borderwidth=25,font=
('calibre', 15, 'bold','underline'))
personal_label = tkinter.Label(top, text = 'Personal Information :
        ',borderwidth=25,font=('calibre', 10, 'bold','underline'))

# creating a label and entry for name using widget Label and Entry rspt.
name_label = tkinter.Label(top, text = 'Name :',borderwidth=25,font=('calibre', 10, 'bold'))

name_entry = tkinter.Entry(top, textvariable = name_var,width= 30, font=('calibre',10,'normal
'))

# creating a label for name using widget Label
TypeofRetirement_label = tkinter.Label(top, text = 'Type of Retirement :',borderwidth=25,font
=('calibre', 10, 'bold'))
n = tkinter.StringVar()
TypeofRetirement = ttk.Combobox(top, width = 32, textvariable = n)
# Adding combobox drop down list
TypeofRetirement['value'] = (' Superannuation', ' Volutary')
TypeofRetirement.current(1)

# creating a label for name using widget Label .
DateofBirth_label = tkinter.Label(top, text = 'Date of Birth :',borderwidth=25,font=('calibre
', 10, 'bold'))
d = tkinter.StringVar()
m = tkinter.StringVar()
y = tkinter.StringVar()
Date = ttk.Combobox(top, width = 5, textvariable = d)
# Adding combobox drop down list
Date['value'] = ('Date','1', '2', '3', '4', '5', '6', '7', '8', '9', '10',
        '11', '12', '13', '14', '15', '16', '17', '18', '19', '20',
        '21', '22', '23', '24', '25', '26', '27', '28', '29', '30', '31')
Date.current(0)
Month = ttk.Combobox(top, width = 10, textvariable = m,)
# Adding combobox drop down list
Month['value'] = ('Month','January', 'February', 'March', 'April', 'May', 'June', 'July', 'Au
gust', 'September', 'October', 'November', 'December')
Month.current(0)
Year = ttk.Combobox(top, width = 7, textvariable = y)
# Adding combobox drop down list
Year['value'] = ('Year','2020','2019','2018','2017','2016','2015','2014','2013','2012','2011'
,'2010',
        '2009','2008','2007','2006','2005','2004','2003','2002','2001','2000'
,'1999',
        '1998','1997','1996','1995','1994','1993','1992','1991','1990','1989'
,'1988',
        '1987','1986','1985','1984','1983','1982','1981','1980','1979','1978'
,'1977',
        '1976','1975','1974','1973','1972','1971','1970','1969','1968','1967'
,'1966',

```

```

        '1965','1964','1963','1962','1961','1960','1959','1958','1957','1956'
    , '1955',
        '1954','1953','1952','1951','1950','1949','1948','1947','1946','1945'
    , '1944',
        '1943','1942','1941','1940')
Year.current(0)

# creating a label for name using widget Label
DateofRetirement_label = tkinter.Label(top, text = 'Date of Retirement :',borderwidth=25,font
=('calibre', 10, 'bold'))
rd = tkinter.StringVar()
rm = tkinter.StringVar()
ry = tkinter.StringVar()
rDate = ttk.Combobox(top, width = 5, textvariable = rd)
# Adding combobox drop down list
rDate['value'] = ('Date','1', '2', '3', '4', '5', '6', '7', '8', '9', '10',
                '11', '12', '13', '14', '15', '16', '17', '18', '19', '20',
                '21', '22', '23', '24', '25', '26', '27', '28', '29', '30', '31')
rDate.current(0)
rMonth = ttk.Combobox(top, width = 10, textvariable = rm,)
# Adding combobox drop down list
rMonth['value'] = ('Month','January', 'February', 'March', 'April', 'May', 'June', 'July', 'A
ugust', 'September', 'October', 'November', 'December')
rMonth.current(0)
rYear = ttk.Combobox(top, width = 7, textvariable = ry)
# Adding combobox drop down list
rYear['value'] = ('Year','2020','2019','2018','2017','2016','2015','2014','2013','2012','2011
','2010',
                '2009','2008','2007','2006','2005','2004','2003','2002','2001','2000
','1999',
                '1998','1997','1996','1995','1994','1993','1992','1991','1990','1989
','1988',
                '1987','1986','1985','1984','1983','1982','1981','1980','1979','1978
','1977',
                '1976','1975','1974','1973','1972','1971','1970','1969','1968','1967
','1966',
                '1965','1964','1963','1962','1961','1960','1959','1958','1957','1956
','1955',
                '1954','1953','1952','1951','1950','1949','1948','1947','1946','1945
','1944',
                '1943','1942','1941','1940')
rYear.current(0)

# creating a label and entry for name using widget Label and Entry rspt.
input_label = tkinter.Label(top, text = ' Input to calculate pension :
                ',borderwidth=25,font=('calibre', 10, 'bold','underline'))
TotalQualifyingService_label = tkinter.Label(top, text = 'Total Qualifying Service(in year) :
','borderwidth=25,font=('calibre', 10, 'bold'))
TotalQualifyingServicemonth_entry = tkinter.Entry(top,textvariable = year_var,width= 30, font
=('calibre',10,'normal'))

```

```

# creating a label and entry for name using widget Label and Entry rspt.
explanation_one = '''Sum of Last 10
months Emoluments
(Basic Pay (including Grade Pay)+ NPA)
(in Rs.) :'''
tenmonthsEmoluments_label = tkinter.Label(top, text = explanation_one ,borderwidth=25,font=('calibre', 10, 'bold'))
tenmonthsEmoluments_entry = tkinter.Entry(top,textvariable = ten_var,width=30, font=('calibre',10,'normal'))

# creating a label and entry for name using widget Label and Entry rspt.
explanation_two = '''Sum of Last
month Emoluments
(Basic Pay (including Grade Pay)+ NPA)
(in Rs.) :'''
onemonthEmoluments_label = tkinter.Label(top, text = explanation_two ,borderwidth=25,font=('calibre', 10, 'bold'))
onemonthEmoluments_entry = tkinter.Entry(top,textvariable = one_var,width=30, font=('calibre',10,'normal'))

# Create a Button
def CalcBasicPension():
    temp1 =one_var.get()
    temp2 = year_var.get()
    basic_pension=(temp1*temp2)/7
    basic_pension2 = basic_pension
    messagebox.showinfo("Your Basic Pension is ",basic_pension)
checkbtn = tkinter.Button(top, text = 'Check my Basic pension',width= 30, bd = '4', command =
    CalcBasicPension)

def commutation_Calc_redirect():
    os.system('python Commutation.py')
nextbtn = tkinter.Button(top, text = 'Check Your Commutation Now!',width= 30, bd = '4', command =
    commutation_Calc_redirect)

Quitbtn = tkinter.Button(top, text = 'Quit',width= 15, bd = '4', command = quit)

# placing the label and entry in the required position using place method
title_label.place(x=300, y=5)
personal_label.place(x=60 ,y=55 )
name_label.place(x=100, y=100)
name_entry.place(x=250, y=125)
DateofBirth_label.place(x=100, y=150)
Date.place(x=250, y=170)
Month.place(x=310, y=170)
Year.place(x=400, y=170)
TypeofRetirement_label.place(x=500, y=100)

```

```

TypeofRetirement.place(x=680, y=125)
DateofRetirement_label.place(x=500, y=145)
rDate.place(x=680, y=170)
rMonth.place(x=740, y=170)
rYear.place(x=830, y=170)
input_label.place(x=60,y=220)
TotalQualifyingService_label.place(x=100, y=270)
TotalQualifyingServicemonth_entry.place(x=390, y=295)
tenmonthsEmoluments_label.place(x=100, y=310)
tenmonthsEmoluments_entry.place(x=390, y=365)
onemonthEmoluments_label.place(x=100, y=400)
onemonthEmoluments_entry.place(x=392, y=450)
Quitbtn.place(x=800, y=25)
checkbtn.place(x=250, y=530)
nextbtn.place(x=500,y=530)
top.mainloop()

```

2. Commutation Calculator:

```

# importing tkinter module.
import tkinter
from tkinter import ttk
from PIL import ImageTk,Image
from tkinter import messagebox #for pop-up window
import mysql.connector as sql #to connect to MySQL server
import tkinter.font as font

import os
top = tkinter.Tk()

# Give basic characteristics and geometry to the window.
top.geometry("1000x600")
image = Image.open('LoginImage.png')
#photo_image = ImageTk.PhotoImage(image)
#label = tkinter.Label(top, image = photo_image)
#label.pack()
top.title(" - Pension and Commutation Calculator - ")

name_var=tkinter.StringVar()
year_var=tkinter.IntVar()
month_var=tkinter.IntVar()
one_var=tkinter.IntVar()
ten_var=tkinter.IntVar()
commutation_var=tkinter.IntVar()
basic_pension= tkinter.IntVar()
reduced_pension= tkinter.IntVar()
totalcommutation_var= tkinter.IntVar()

```

```

def calcom():
    temp2=commutation_var.get()
    temp3=basic_pension.get()
    if(temp2<=33):
        temp=((temp2/100)*temp3)*10
        totalcommutation_var=temp
        temp1=(temp3 - (temp2/100)*temp3)
        reduced_pension= temp1
        messagebox.showinfo("Total commutation is
        ", totalcommutation_var)
        messagebox.showinfo("Reduced pension is
        ", reduced_pension)
    else:
        messagebox.showinfo("Warning!! ", "invalid commutation value. Please enter a valid va
        lue less than 33")

def Redirect_Basic_Pension():
    os.system('python BAsic_Pension.py')

# creating a label and entry for name using widget Label and Entry rspt
title_label = tkinter.Label(top, text = ' --- Commutation Calculator --
- ',borderwidth=25,font=('calibre', 15, 'bold','underline'))
basicpension_label = tkinter.Label(top, text = 'Your Basic Pension :',borderwidth=5,font=('ca
libre', 10, 'bold'))
basicpension_entry = tkinter.Entry(top,textvariable = basic_pension,width=30, font=('calibre'
,10,'normal'))
Calc_Basic_Pension_label = tkinter.Label(top, text = 'If you have not Calculated your Basic P
ension --->',borderwidth=25,font=('calibre', 10, 'bold','underline'))
Calc_Basic_Pension_btn = tkinter.Button(top, text = 'Calculate your Basic Pension Here',width
= 40, bd = '4', command = Redirect_Basic_Pension)
commutation_label = tkinter.Label(top, text = 'Commutation(Maximum 33%) :',borderwidth=5,font
=('calibre', 10, 'bold'))
commutation_entry = tkinter.Entry(top,textvariable = commutation_var,width=30, font=('calibre
',10,'normal'))

calCommutation_btn = tkinter.Button(top, text = 'Calculate my Commutation and Final Pension',
width= 40, bd = '4', command = calcom)

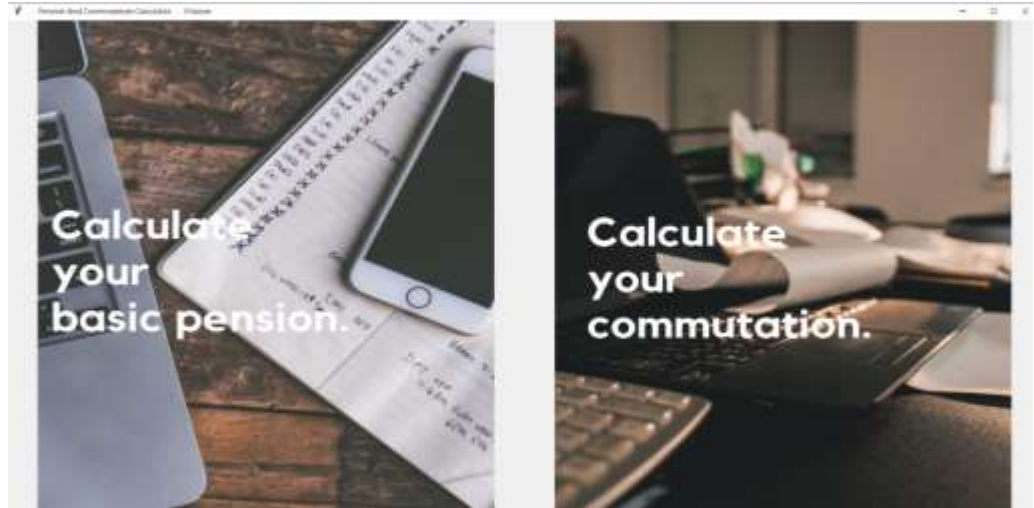
# placing the label and entry in the required position using place method

title_label.place(x=250,y=100)
basicpension_label.place(x=120, y=300)
basicpension_entry.place(x=330, y=300)
commutation_label.place(x=120, y=400)
commutation_entry.place(x=330, y=400)
calCommutation_btn.place(x=630,y=500)
Calc_Basic_Pension_btn.place(x=500,y=350)
Calc_Basic_Pension_label.place(x=500,y=250)
top.mainloop()

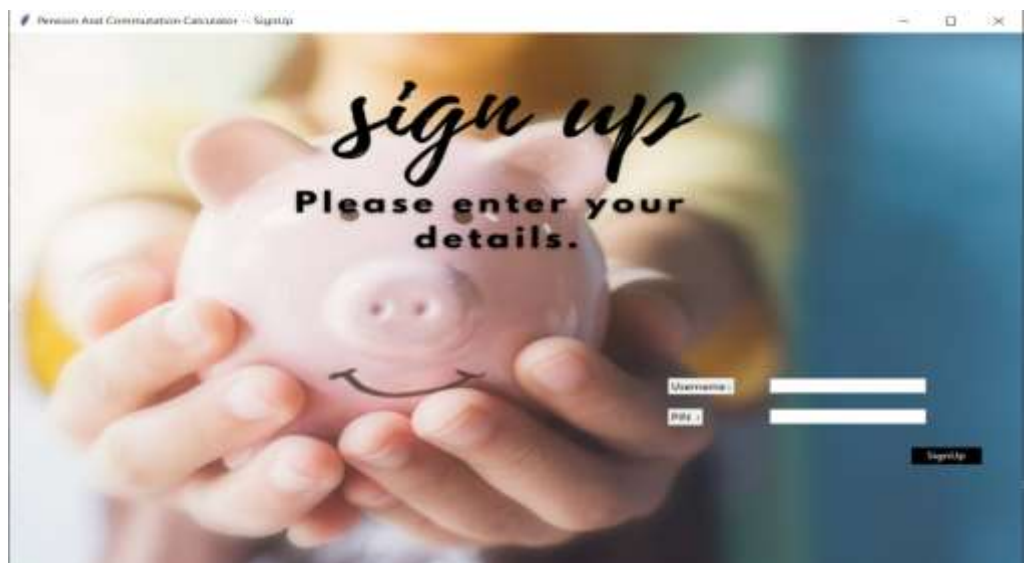
```


SCREENSHOT

Calculator
Choosing
Window



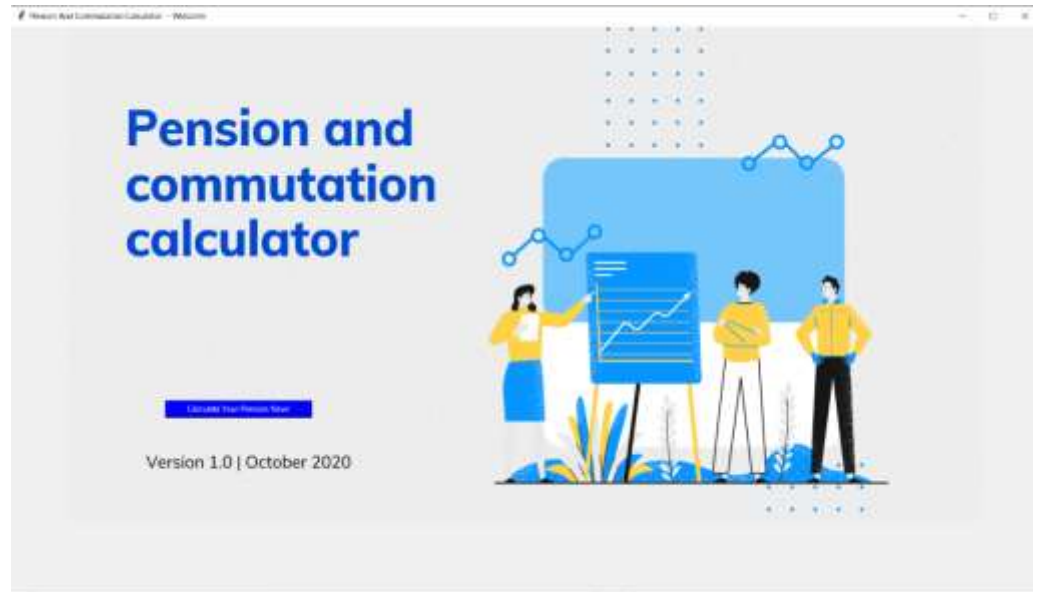
Sign Up
Page



Login
Page



Welcome Page



Basic Pension Calculator

BASIC PENSION CALCULATOR

Personal Information :

Name : Type of Retirement :

Date of Birth : Date of Retirement :

Input to calculate pension :

Total Qualifying Service(in year) :

Sum of Last 10 months Emoluments (Basic Pay (including Grade Pay)+ NPA) (in Rs.) :

Sum of Last month Emoluments (Basic Pay (including Grade Pay)+ NPA) (in Rs.) :

Basic Pension Result

BASIC PENSION CALCULATOR

Personal Information :

Name : Type of Retirement :

Date of Birth : Date of Retirement :

Input to calculate pension :

Total Qualifying Service(in year) :

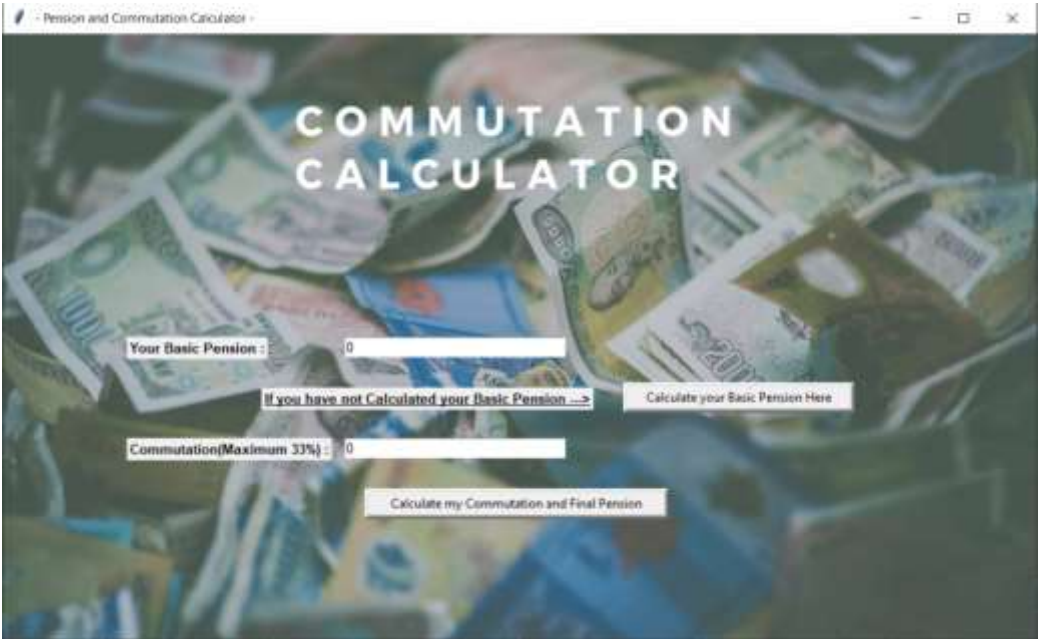
Sum of Last 10 months Emoluments (Basic Pay (including Grade Pay)+ NPA) (in Rs.) :

Sum of Last month Emoluments (Basic Pay (including Grade Pay)+ NPA) (in Rs.) :

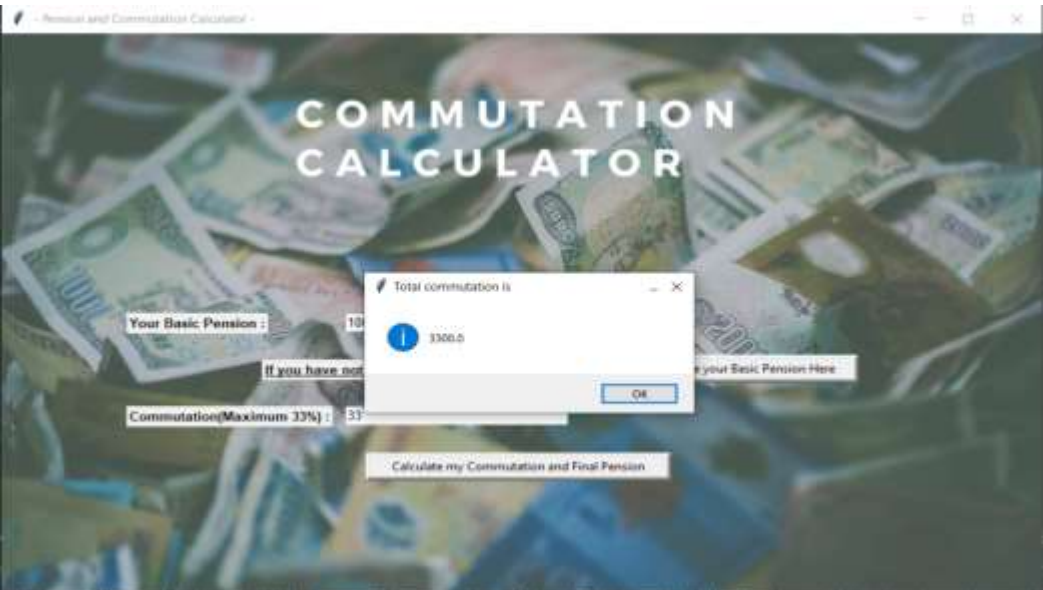
Your Basic Pension is

21426.571428571428

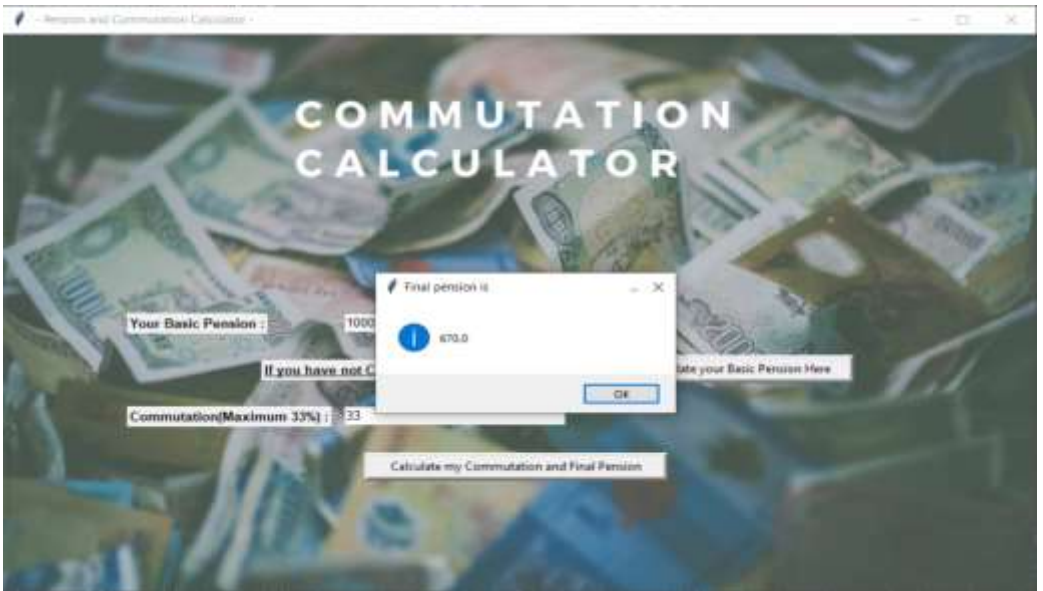
Commu
tation
Calculat
or



Commuta
tion
Result



Resuced
Pension
Result





CONCLUSION

We finally got an end project “PENSION AND COMMUTATION CALCULATOR” that includes all the modules that are above mentioned. We have learnt how to make a GUI using tinker in python and also learnt database connectivity using MySQL. This calculator offers the number of tools which helps you determine everything from your monthly pension/basic pension, commutation per year to total commutation. Each of those tools will help any individual to easily access and know about their pension and commutation.

