APPENDIX 1

Basic Pension Amount Payable Calculator for Bank Employees (Calculate Pension Payable & Commutation and Calculate Gratuity (Max 20 Lacs))

END TERM REPORT

By

Roshan Dhaker (11906174),

Ayush Chourasia (11906115),

Macha Khushi Reddy (11906139).

Section: **K19PG**

Roll Numbers: **23, 25, 38**

Department of Intelligent Systems,

School of Computer Science Engineering,

Lovely Professional University, Jalandhar

November, 2020

Student Declaration

This is to declare that this report has been written by us. No part of the report is copied from other sources. All information included from other sources have been duly acknowledged. We aver that if any part of the report is found to be copied, We are shall take full responsibility for it.

Signature: M. Khushi Reddy, Ayush Chourasia,
Roshan Dhaker.
Name: M. Khushi Reddy, Ayush Chourasia,
Roshan Dhaker.
Roll Number: 38, 25, 23.

Place: Lovely Professional University, Jalandhar.

Date: 31st Oct, 2020

TABLE OF CONTENTS

TITLE **PAGENO.**

BONAFIDE CERTIFICATE

Certified that this project report "Basic Pension Amount Payable Calculator for Bank Employees" is the bonafide work of **"Khushi Reddy (11906139), Ayush Chourasia (11906115), Roshan Dhaker (11906174),**" who carried out the project work under my supervision.

<<Signature of the Supervisor>> (exempted)

Dr. Dhanpratap Singh

Associate Professor

25706

Intelligence System

BACKGROUND

Almost each and every person in today’s world thinks about saving their money for future. Everybody wants to save money so that they can live a peaceful life in the future.

And due to this, everybody wants to calculate how much money they will get once they leave their jobs which is known as pension. And to be very honest, it is not very easy to calculate the pension. You will need to do a lot of calculations, a lot of time and obviously a good brain to perform the calculations. And if you did a mistake while calculating, all the work goes in vain and there is no way to rectify the mistake until you are able to find the root of the mistake.

So, in today’s modern world where machines are replacing humans for working, why not make an application which will reduce human efforts and time along with no errors to calculate the pensions within milliseconds. The application will be known as ‘Pension Calculator’ and will be super-efficient and very easy to use. It will be an open-source program without any cost and will be easily accessible for everyone. It will be available on different platforms like Android, iOS, Windows, Linux, and much more.

At the beginning, we will release the windows version but later, it will be available on all the Mentioned platforms. Also, the program will be regularly maintained by the developer’s team And any error or disfunction will be properly monitored by the team. Also, all the data Entered by the user will be protected by our policy service. You can read the other information about policies on our website.

DESCRIPTION OF PROJECT

There are many graphical user interface (GUI) toolkits that we can use with the Python programming language. The big three are Tkinter, wxPython, and PyQt. Each of these toolkits will work with Windows, macOS, and Linux, with PyQt having the additional capability of working on mobile. And for this project we have used Tkinter one of toolkits of GUI.

A graphical user interface is an application that has buttons, windows, and lots of other widgets that the user can use to interact with your application. A good example would be a web browser. It has buttons, tabs, and a main window where all the content loads.

Tkinter is a Python Package for creating GUI applications. Python has a lot of GUI frameworks, but Tkinter is the only framework that's built into the Python standard library. Tkinter has several strengths; it's cross-platform, so the same code works on Windows, macOS, and Linux. Tkinter is lightweight and relatively painless to use compared to other frameworks. This makes it a compelling choice for building GUI applications in Python, especially for applications where a modern shine is unnecessary, and the top priority is to build something that's functional and cross-platform quickly.

So as mentioned above, a graphical user interface (GUI) is an interface that is drawn on the screen for the user to interact with. User interfaces have some common components:

- Main window
- Menu
- Toolbar
- Buttons
- Text Entry
- Labels

All of these items are known generically as **widgets**. There are many other common widgets and many custom widgets that Tkinter supports. A developer will take the widgets and arrange them logically on a window for the user to interact with.

Creating the Skeleton of Application

An application skeleton in a GUI context is a user interface with widgets that don't have any event handlers. These are useful for prototyping. You basically just create the GUI and present it to your concerned head for sign-off before spending a lot of time on the backend logic.

from tkinter import *

```
window = Tk()

window.geometry("800x600")

window.mainloop()
```

In above example, you have two parts: **window** and the **mainloop**. The **window** is Tkinter’s application object and is required for running your GUI. The **window** starts something called **window.mainloop()**. This is the event loop that is required to run an application.

By default, a **window** will include minimize, maximize, and exit buttons along the top. You won’t normally create an application in this manner though. Most Tkinter code will require you to subclass the window and other widgets so that you can get the full power of the toolkit.

We’ll add a label using the Label Class and change its text configuration as desired. The **grid()** function is a geometry manager which keeps the label in the desired location inside the window. If no parameters are mentioned by default it will place it in the empty cell; that is 0,0 as that is the first location.
Now add a button to the root window. Changing the button configurations gives us a lot of options. In this example we will make the button display a text once it is clicked and also change the color of the text inside the button.
Using the **Entry()** class we will create a text box for user input. To display the user input text, we’ll make changes to the function **clicked()**. We can get the user entered text using the **get()** function. When the **Button** after entering of the text, a default text concatenated with the user text. Also change button grid location to column 2 as **Entry()** will be column 1.
To add a menu bar, you can use **Menu** class. First, we create a menu, then we add our first label, and finally, we assign the menu to our window. We can add menu items under any menu by using **add_cascade().**

Below is the Example that has been explained above,

```
from tkinter import *
root = Tk()
root.title("Welcome to GeekForGeeks")
root.geometry('350x200')
menu = Menu(root)
item = Menu(menu)
item.add_command(label='New')
menu.add_cascade(label='File', menu=item)
root.config(menu=menu)
lbl = Label(root, text = "Are you a Geek?")
```

```
lbl.grid()
txt = Entry(root, width=10)
txt.grid(column =1, row =0)
def clicked():

res = "You wrote" + txt.get()
lbl.configure(text = res)
btn = Button(root, text = "Click me" ,
fg = "red", command=clicked)
btn.grid(column=2, row=0)

root.mainloop()
```

The Components/Widgets used in the Project are:

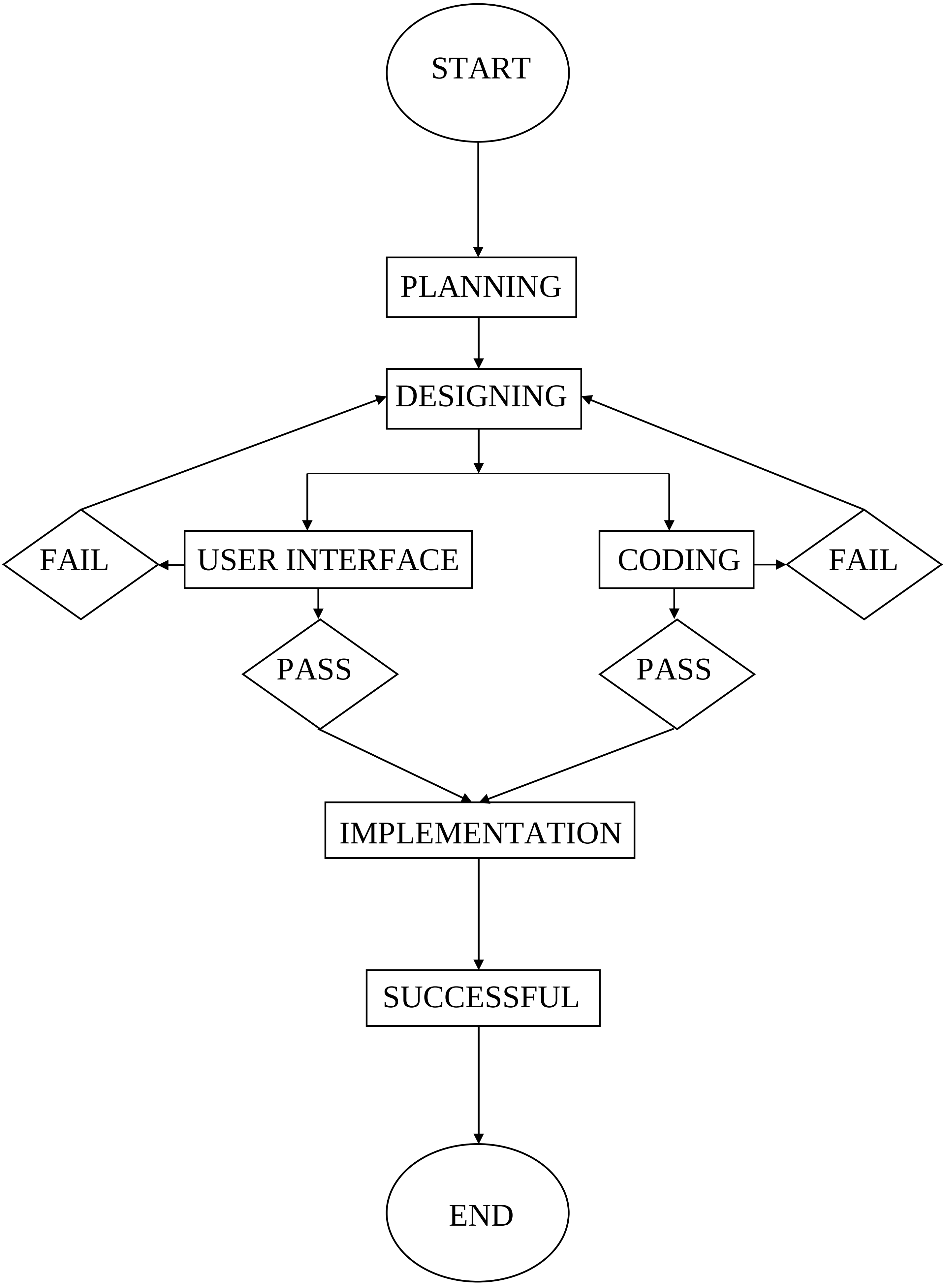
Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called **Widgets.** The list of commonly used **Widgets** are mentioned below:

LABLE: The Label widget is used to provide a single-line caption for other widgets. It can also contain images.
BUTTON: The Button widget is used to display buttons in your application.
ENTRY: The Entry widget is used to display a single-line text field for accepting values from a user.
CHECKBUTTON: The Checkbutton widget is used to display a number of options as checkboxes. The user can select multiple options at a time.
RADIOBUTTON: The Radiobutton widget is used to display a number of options as radio buttons. The user can select only one option at a time.
LABELFRAME: A labelframe is a simple container widget. Its primary purpose is to act as a spacer or container for complex window layouts.

The Geometry Manager used in the Project are:

All Tkinter widgets have access to specific geometry management methods, which have the purpose of organizing widgets throughout the parent widget area. Tkinter exposes the following geometry manager classes: pack, grid, and place. Their description is mentioned below:

PACK(): This geometry manager organizes widgets in blocks before placing them in the parent widget.
GRID(): This geometry manager organizes widgets in a table-like structure in the parent widget.
PLACE(): This geometry manager organizes widgets by placing them in a specific position in the parent widget.


START
PLANNING
DESIGNING
FAIL
USER INTERFACE
CODING
FAIL
PASS
PASS
IMPLEMENTATION
SUCCESSFUL
END

DESCRIPTION OF WORK DIVISION IN TERMS OF ROLES AMONG STUDENTS

So basically, we need to make a program which will perform 3 different tasks and we have a group of 3 students, so we divided the work among us and each student did one task. The following was the task done by the group members :-

(a) Gratuity Calculator – Khushi Reddy (R.no- 38)

(b) Pension Calculator – Ayush Chourasia (R.no- 25)

(c) Commutation Calculator – Roshan Dhakar (R.no- 23)

Then all the tasks were merged into a single program by Khushi Reddy and the GUI interface Was decided by all the students. The final program was tested by all the students using different values to make sure that no error is present. The report work was also divided equally among all of us and was cross checked by all the members.

IMPLEMENTATION OF SCHEDULED WORK OF PROJECT

First and important thing is to plan the project according to the group members and the topic that has been given, and after you have carefully planned your project, you will be ready to start the project implementation phase, the second phase of the project.

The implementation phase involves putting the project plan into action. It's here that the project manager will coordinate and direct project resources to meet the objectives of the project plan. As the project unfolds, it's the project manager's job to direct and manage each activity to the group members, every step of the way. That's what happens in the implementation phase of the project life cycle: we follow the plan we've put together and handle any problems that come up.

The implementation phase is where you and your project team actually do the project work to produce the deliverables. The word "deliverable" means anything your project delivers. The deliverables for your project include all of the products or services that you and your team are performing for the university, including all the project management documents that you put together.

Most often, changes are identified by looking at performance and quality control data. Routine performance and quality control measurements should be evaluated on a regular basis throughout the implementation phase. Gathering reports on those measurements will help you determine where the problem is and recommend changes to fix it.

When you find a problem, you can't just make a change, because it may be too too long to do. You will need to look at how it affects the double constraint (time, scope) and how it impacts project quality. You will then have to figure out if it is worth making the change. If you evaluate the impact of the change and find that it won't have an impact on the project double constraint, then you can make the change without going through change control. Change control is a set of procedures that lets you make changes in an organized way.

EXPLAINATION

The project has been made accordingly, As soon as you run the code it will open a **home page** where we have used the notebook widget and it consists of 5 different tabs.

1) HOME
2) PENSION
3) COMMUTATION
4) GRATUITY
5) COMMUTATION FACTOR

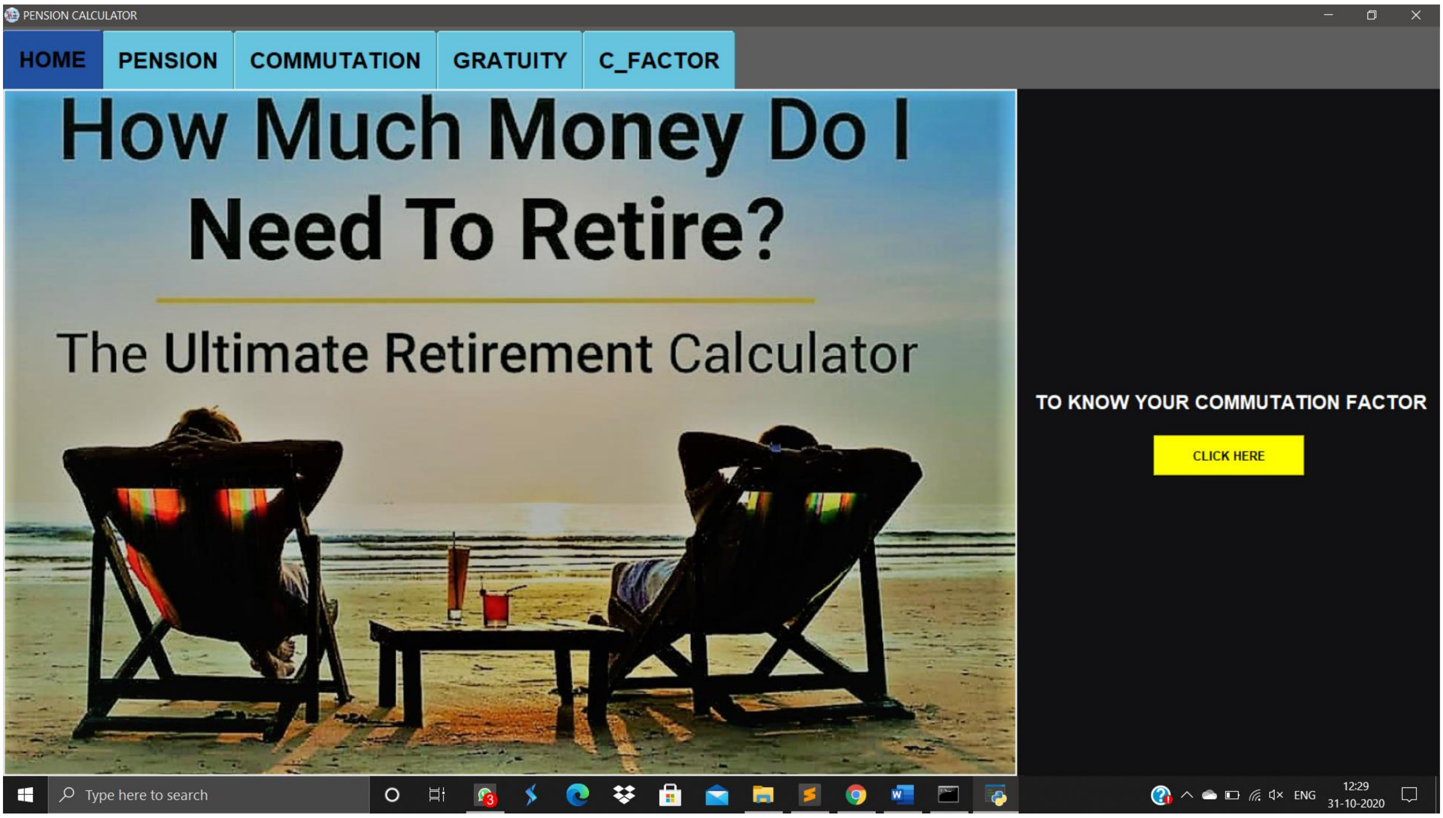
On the home page there is a background image on the frame and also a background color is used.

These are the Four different color's hex codes that have been used in the whole project:

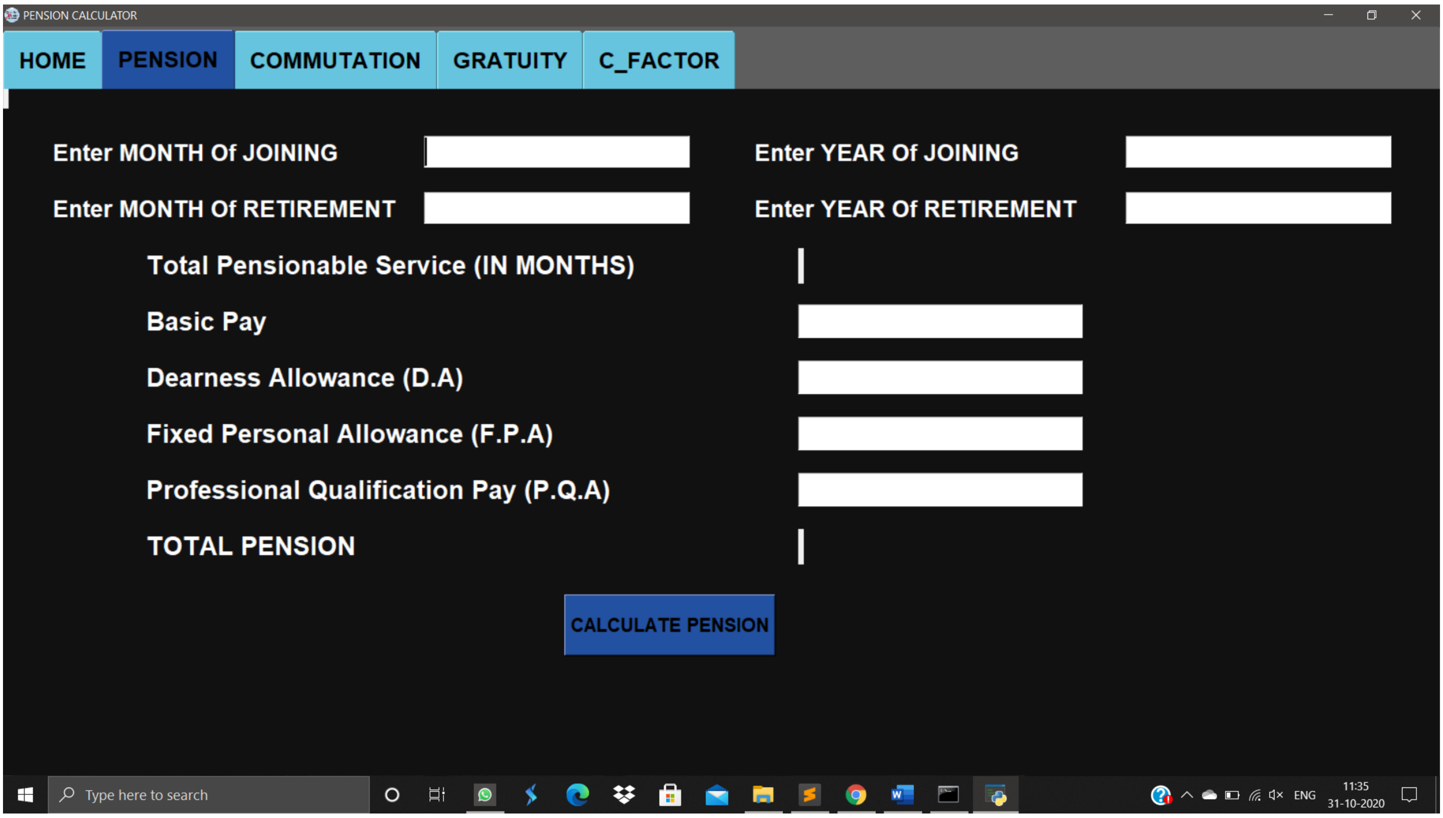
#66c4de for the tab base, #2351a1 for onclick of the tab, #111112 this for the whole notebook frame background, #5e5e5e and this is for the rest of the window. And inside the notebook widgets in each tab we have created a frame that would act as a container for other widgets and in further tabs there it contains a form like structure which the user has to fill to calculate whatever value he or she wants.

IMAGES

1) HOME PAGE (TAB1)



2) PENSION CALCULATOR (TAB2)



Enter MONTH Of JOINING

Enter YEAR Of JOINING

Enter MONTH Of RETIREMENT

Enter YEAR Of RETIREMENT

Total Pensionable Service (IN MONTHS)

Basic Pay

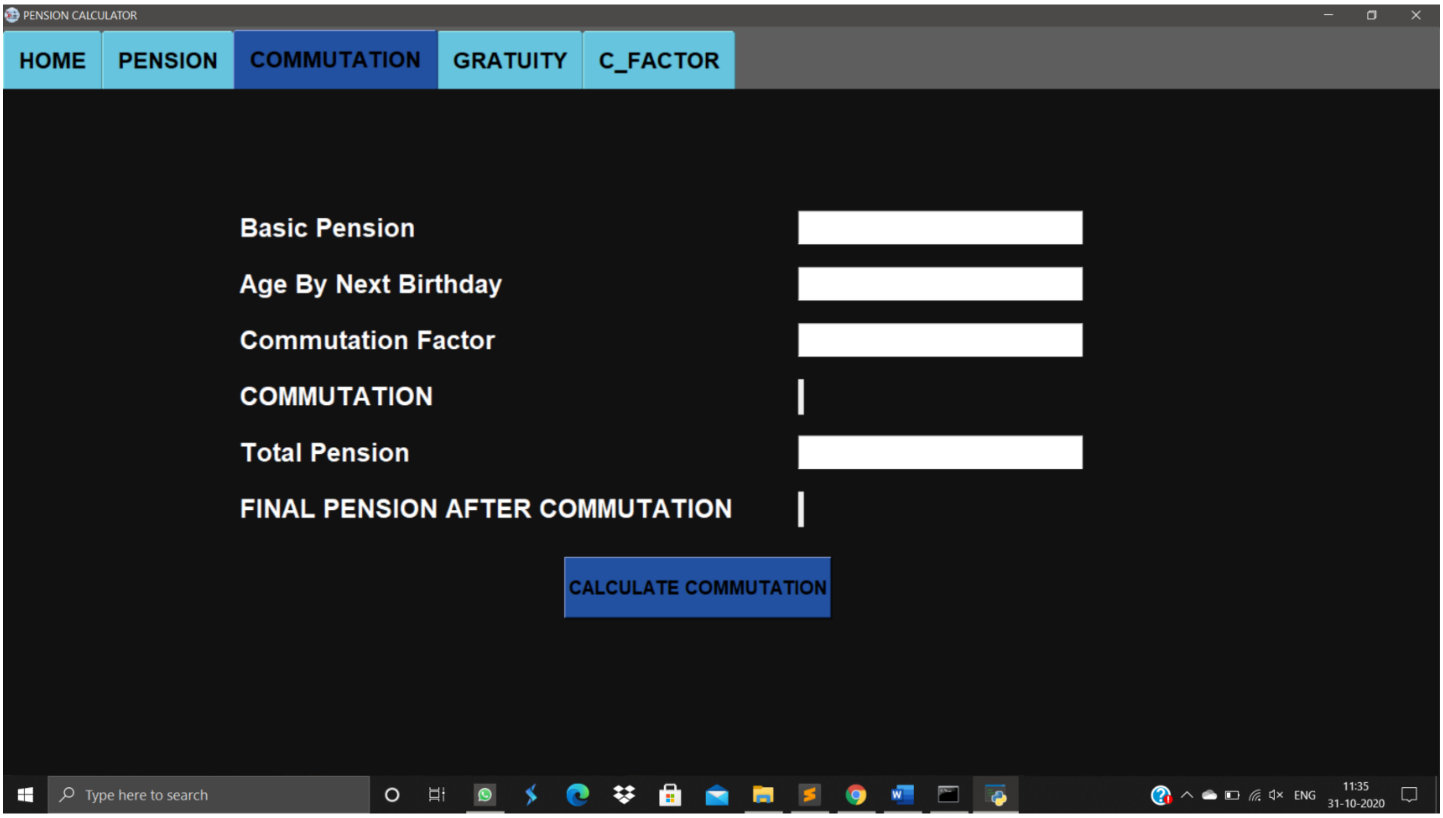
Dearness Allowance (D.A)

Fixed Personal Allowance (F.P.A)

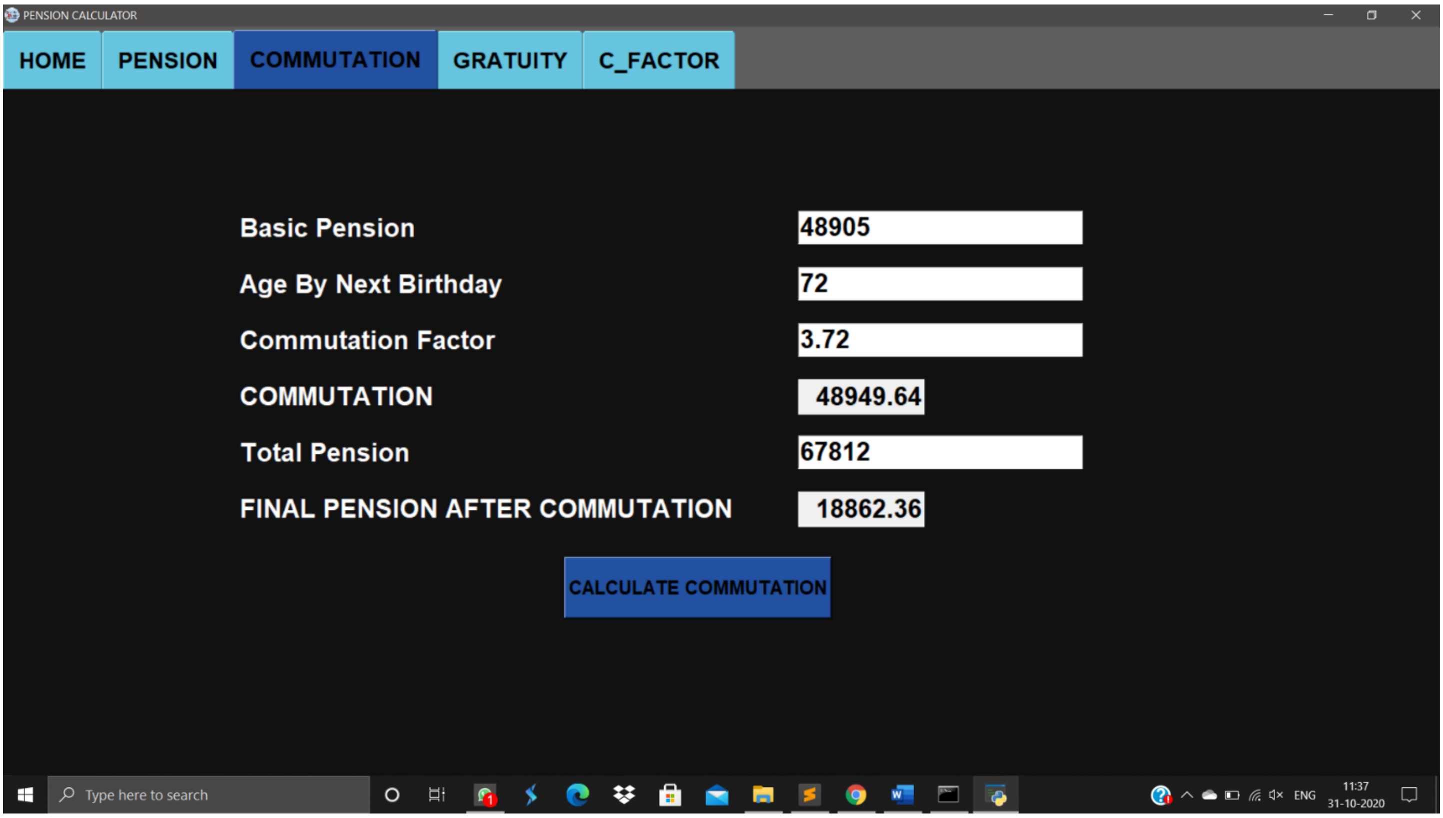
Professional Qualification Pay (P.Q.A)

TOTAL PENSION

3) COMMUTATION CALCULATOR (TAB3)



4) APPLIED AND TESTED



Basic Pension 48905

Age By Next Birthday 72

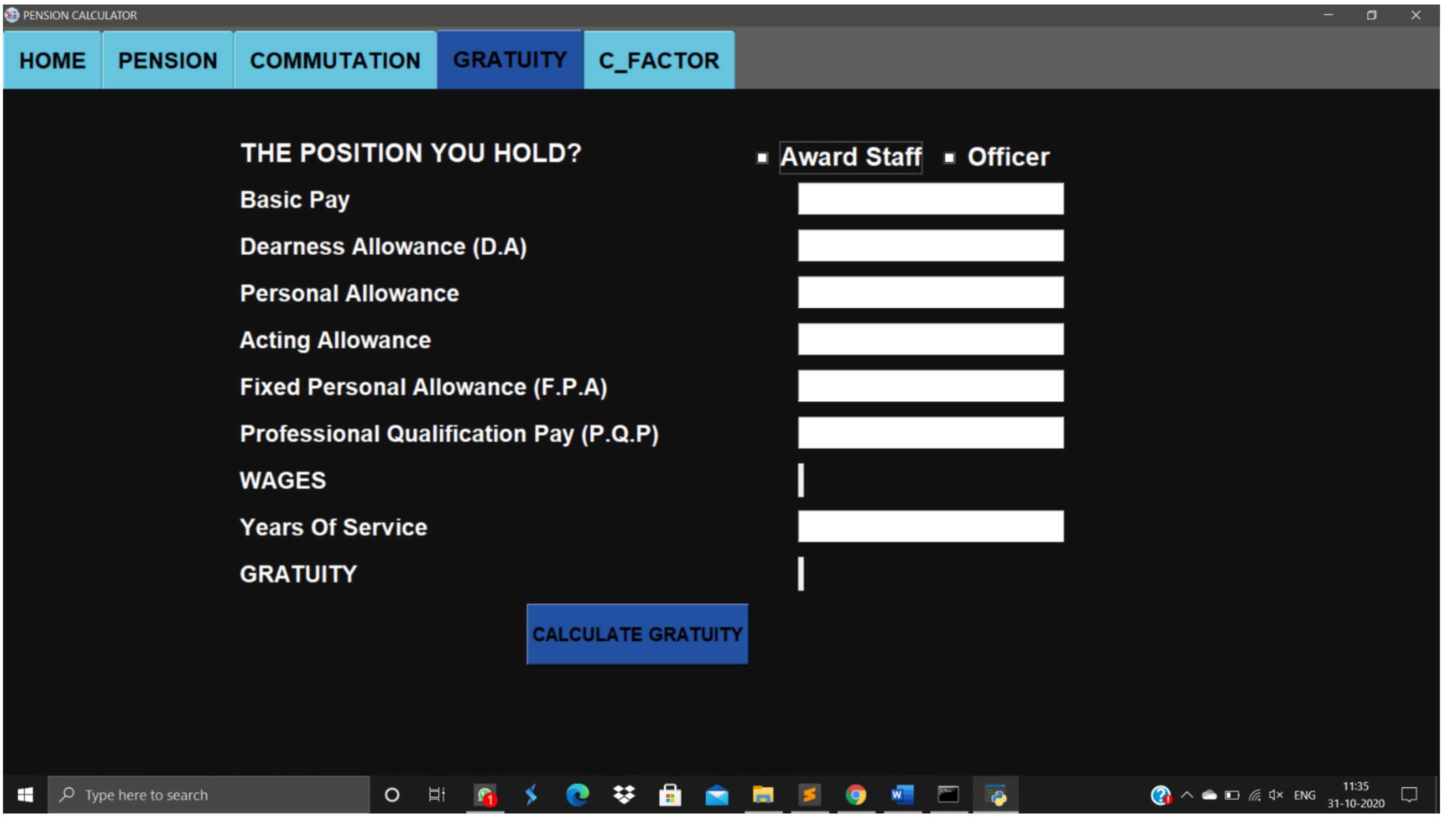
Commutation Factor 3.72

COMMUTATION 48949.64

Total Pension 67812

FINAL PENSION AFTER COMMUTATION 18862.36

5) GRATUITY CALCULATOR (TAB4)



THE POSITION YOU HOLD? ▪ Award Staff ▪ Officer

Basic Pay

Dearness Allowance (D.A)

Personal Allowance

Acting Allowance

Fixed Personal Allowance (F.P.A)

Professional Qualification Pay (P.Q.P)

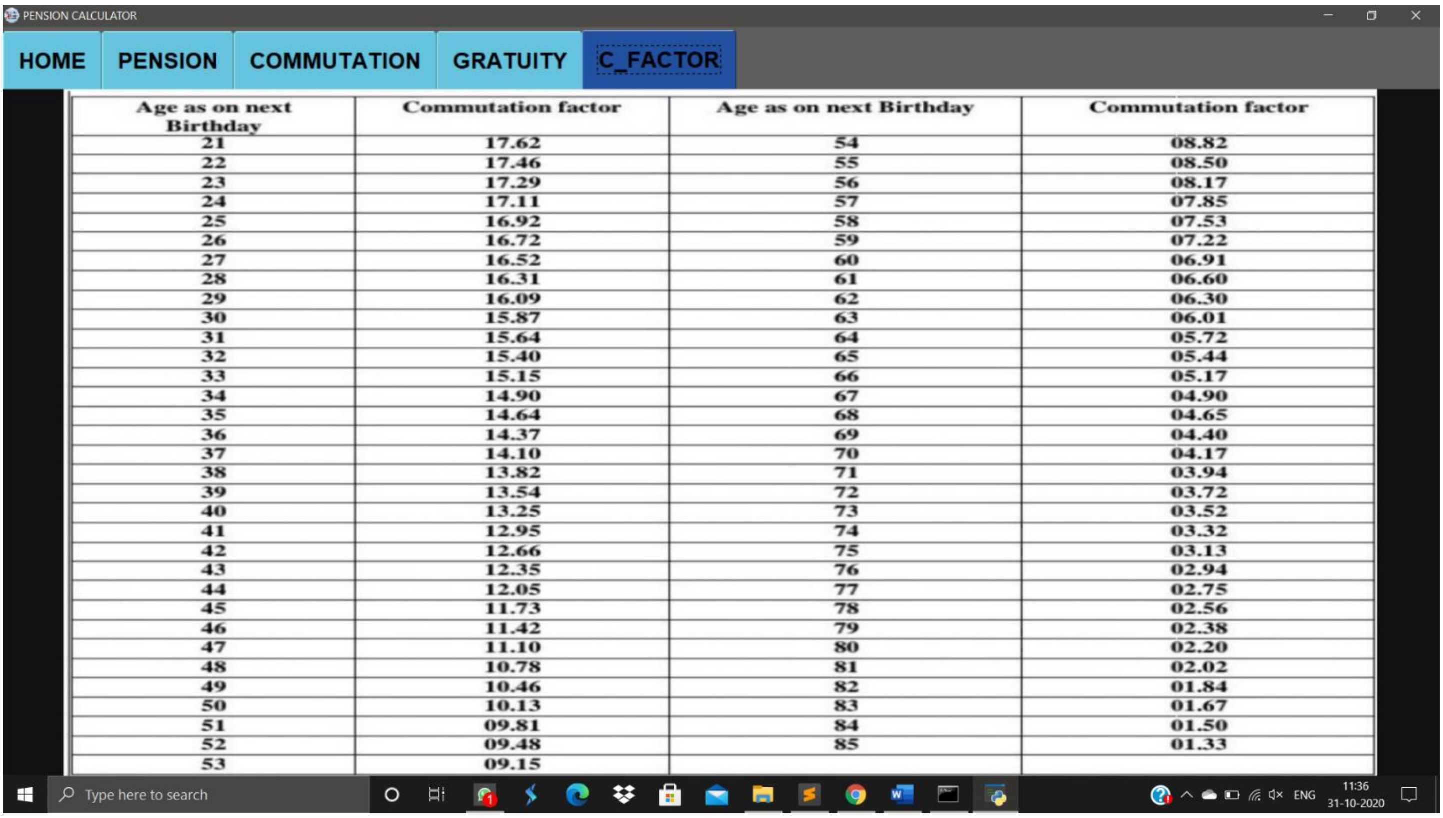
WAGES

Years Of Service

GRATUITY

CALCULATE GRATUITY

6) COMMUTATION FACTORS TABLE (TAB5)



Age as on next Birthday	Commutation factor	Age as on next Birthday	Commutation factor
21	17.62	54	08.82
22	17.46	55	08.50
23	17.29	56	08.17
24	17.11	57	07.85
25	16.92	58	07.53
26	16.72	59	07.22
27	16.52	60	06.91
28	16.31	61	06.60
29	16.09	62	06.30
30	15.87	63	06.01
31	15.64	64	05.72
32	15.40	65	05.44
33	15.15	66	05.17
34	14.90	67	04.90
35	14.64	68	04.65
36	14.37	69	04.40
37	14.10	70	04.17
38	13.82	71	03.94
39	13.54	72	03.72
40	13.25	73	03.52
41	12.95	74	03.32
42	12.66	75	03.13
43	12.35	76	02.94
44	12.05	77	02.75
45	11.73	78	02.56
46	11.42	79	02.38
47	11.10	80	02.20
48	10.78	81	02.02
49	10.46	82	01.84
50	10.13	83	01.67
51	09.81	84	01.50
52	09.48	85	01.33
53	09.15		

TECHNOLOGIES AND FRAMEWORK TO BE USED

The program was built on windows platform and the basic language used in the project is Python. The latest version of the python software is used i.e. Python 3.9
Different members used different Python compilers for the program. Some used the Python Basic software while others used softwares such as Anaconda and PyCharm to complete the Program. However, it was ensured that no compatibility issues will arise while merging the Codes together. Inside Python, the GUI was built by using Tkinter which was imported Into python using code.

SWOT ANALYSIS ACHIEVED IN PROJECT

STRENGTHS

There are a lot of strengths of the program. It helps to easily calculate the pension, Commutation and gratuity and helps in reducing human efforts and decreasing time and reduces the chances of errors. It helps to have a clearer vision over the future of a person.

WEAKNESS

Everything comes with some drawbacks and hence this program also comes with some Weakness. The program doesn’t provide users to save their details and other information So that they can reuse it when they will use the application once again. However, our Team is working on this and soon we will introduce this feature in our application.

OPPORTUNITIES

We as a developer of the application are looking for a lot of opportunities awaiting For us in near future. Since this is a very helpful application, we are looking to release it globally as soon as possible so that people all over the world can avail the features of this application for their benefits. Also, we are looking for some government support so that we can make this application officially approved by the government and easily available to each and every one through our government.

THREATS

We are monitoring the application very carefully, however there are threats to the software that we would like to avoid for the smooth running of the software. The hackers may be targeting our servers to stop the application and to steal the user data available on the servers. Also, since it is an open-source software, may be someone will use it to earn money by modifying the application and then sell it to users who may be unaware of our application which is totally free. So, it is advisable to share the application as much as a user can so that everyone will come to know about our software and will be able to use it for free.

GITHUB ACCOUNT LINK

https://github.com/khushireddy05

NAME OF THE REPOSITORY: PYTHON_PROJECT