

# Content

1. <i>Objective</i> .....	4
2. <i>Introduction</i> .....	4
3. <i>Design of Project</i> .....	7
4. <i>Table used &amp; flow chart</i> .....	8
5. <i>Result screenshot</i> .....	10
6. Source code .....	13
7. <i>Conclusion</i> .....	29
8. <i>References</i> .....	30

# Objective

For college students keeping track of their CGPA is important so that they can keep a balance between their academics and co-curricular activities along with good marks thus its much difficult to keep on calculating again and again the TGPA's and CGPA and hence this **CGPA calculator** helps to instantly calculates the TGPA and CGPA of a student based on certain inputs. We have created a very user-friendly graphical user interface so that user can easily run through it without any difficulty. If a user wants to see the grade points related to each grade, they can see easily. The major purpose of the project is to help students keep track of their grades and markings. Also, they can predict how much they must score to keep the desired CGPA for the year. Calculation of TGPA'S and CGPA have become much easier through this project the student is easily able to know the grading system.

# Introduction

The project is based on – “CGPA Calculator”. Cumulative Grade Point Average (CGPA) is an educational ranking/evaluation method. The CGPA is a figure that reflects the grade point average for all classes you have taken and for classes for which you have received credit by means such as testing at your school/College/University/School/College/university policies vary in the way they evaluate credit for courses transferred to your current school from another. Your school's registrar will have information about your school's policy.

To calculate your CGPA you need to know the total number of grade points you have earned and the total number of credit hours you have attempted. While countries and institutions use a range of alpha-numerical grading scales, their formulas for calculating a student's cumulative grade point average (CGPA) score are basically the same. In mathematical terms, the CGPA score is a "weighted

mean," wherein the influence each grade has on the cumulative score depends on the number of credit hours the course was worth. Additionally, courses audited or taken on a "Pass/Fail" basis are omitted from the CGPA calculations completely.

In this project, filling the required information for Student Session, Name of Courses, Course Title, Course Code, Credit Unit, Grade Letter then clicking on the "Calculate GPA" button will display the resulting CGPA on the screen.

Formula for calculating the CGPA:

$$\frac{\sum (\text{No.ofCredits} \times \text{GradePoint})}{\sum (\text{No.ofCredits})}$$

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.

The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms and can be freely distributed. Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test debug cycle is incredibly fast. Debugging Python programs is easy a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception.

When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source the fast edit-test-debug cycle makes this simple approach very effective. While The Python Language Reference describes the exact syntax and semantics of the Python

language, this library reference manual describes the standard library that is distributed with Python. It also describes some of the optional components that are commonly included in Python distributions. Python's standard library is very extensive, offering a wide range of facilities as indicated by the long table of contents listed below.

The library contains built-in modules (written in C) that provide access to system functionality such as file I/O that would otherwise be inaccessible to Python programmers, as well as modules written in Python that provide standardized solutions for many problems that occur in everyday programming. Some of these modules are explicitly designed to encourage and enhance the portability of Python programs by abstracting away platform-specifics into platform-neutral APIs.

The Python installers for the Windows platform usually include the entire standard library and often also include many additional components. For Unix-like operating systems Python is normally provided as a collection of packages, so it may be necessary to use the packaging tools provided with the operating system to obtain some or all the optional components. In addition to the standard library, there is a growing collection of several thousand components (from individual programs and modules to packages and entire application development frameworks), available from the Python Package Index.

tkinter: A GUI toolkit for python.

This whole GUI is made by using Tkinter which is one of the best libraries of Python for making GUI. And it provides each single feature of GUI like use of buttons, checkboxes, entry boxes and labels. Its all commands are very simple and easy to implement

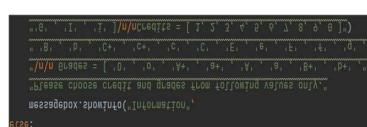
# DESIGN of Program

## overview

1. On the top of the window frame, there is Label consisting of image of Title of program.
2. Various Labels are used in the top, middle and bottom- left position. The 2 TGPA's, CGPA labels will change its texts and show the result of calculated TGPA or CGPA accordingly.
3. There are 16 Entry widgets in each Semester (2) in total 32, if someone enters wrong info it'll show error message, otherwise stores the input for calculation.
4. There are a total of 5 buttons used, two for calculating TGPA's, another one for calculating CGPA, one for clearing all inputs and values, and the last one for getting info about grade points on new window.
  - 4.1. "Grade information" button is placed in the top right corner, when clicked will open the new window with the information about grade point and grade.
  - 4.2. "CLEAR ALL" button is placed in the right corner and will reset the data entered and GPA to its initial state.
5. The dimensions used for the frame are 978x610.

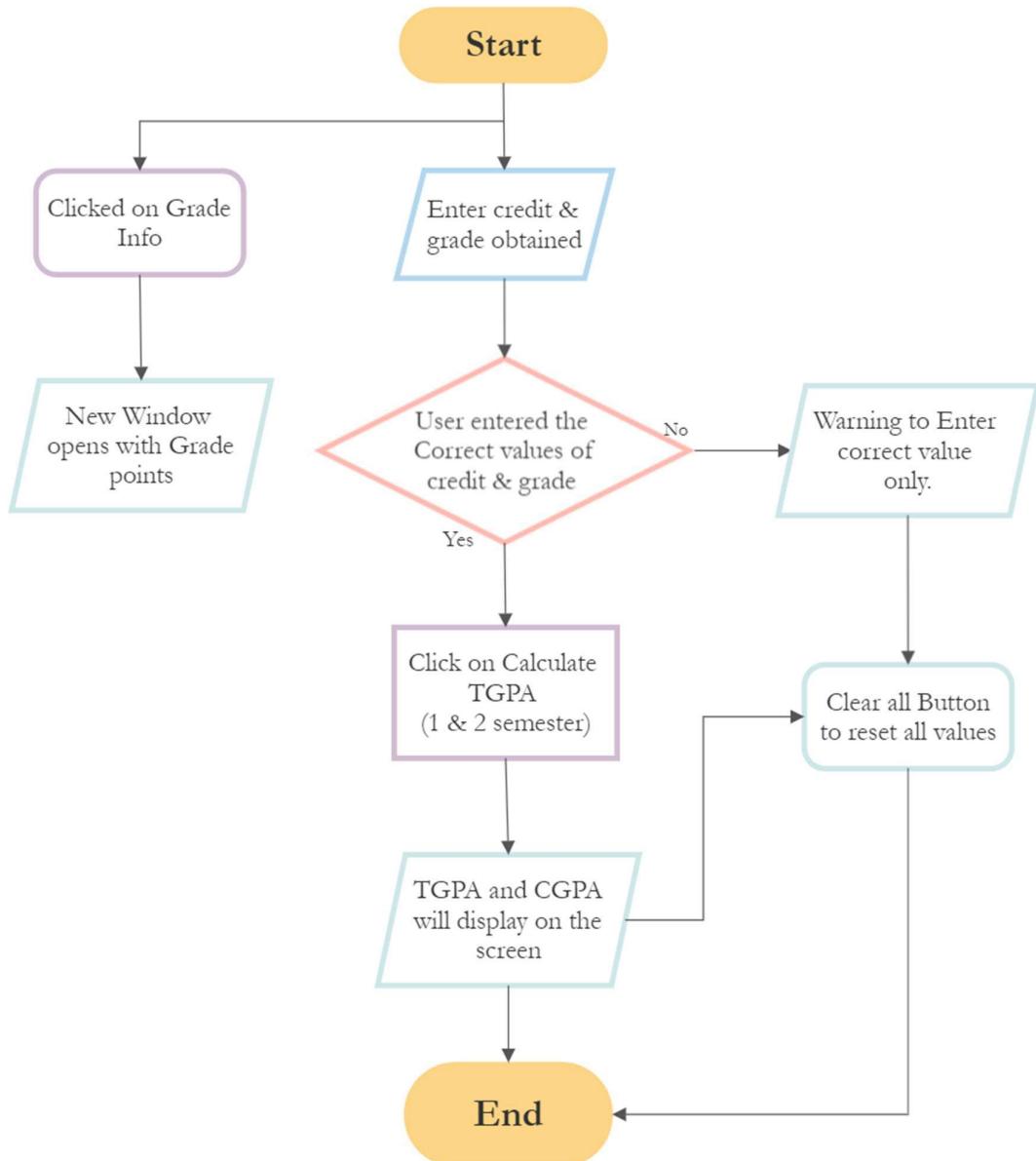
## Table Used

There are various functions in this project to provide the functionality to our project which are given below with the explanation:

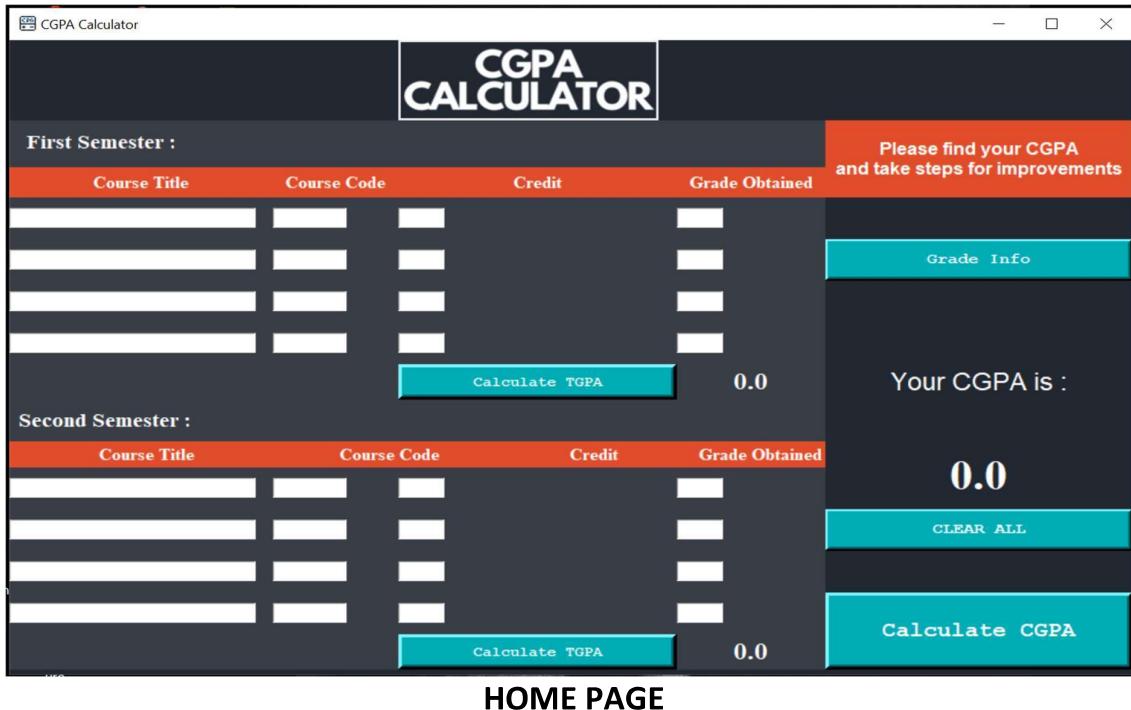
To calculate TGPA of first semester	<pre>i = 0 self.tgpa1, first_part1, second_part1 = 0.00, 0, 0 while i &lt; 4:     first_part1 += credit1[i] * grade1point[i]     second_part1 += credit1[i]     i = i + 1 self.tgpa1 = first_part1 / second_part1 self.tgpa1 = "{:.2f}".format(self.tgpa1) print(self.tgpa1) self.firsttgpavalue["text"] = self.tgpa1</pre>	<p>This function is used to calculate the TGPA of first semester,</p> <ol style="list-style-type: none"> <li>If value entered is incorrect, Error message will come.</li> <li>If all the required values are not entered by the user than exception will be thrown with a message box.</li> </ol>
To calculate TGPA of second semester	<pre>j = 0 self.tgpa2, first_part2, second_part2 = 0.00, 0, 0 while j &lt; 4:     first_part2 += credit2[j] * grade2point[j]     second_part2 += credit2[j]     j = j + 1 self.tgpa2 = first_part2 / second_part2 self.tgpa2 = "{:.2f}".format(self.tgpa2) print(self.tgpa2) self.secondtgpavalue["text"] = self.tgpa2 print("tgpa1=", self.tgpa1) print("tgpa2=", self.tgpa2)</pre>	<p>This function is used to calculate the TGPA of second semester,</p> <ol style="list-style-type: none"> <li>Exception handling</li> </ol> <pre>except:     messagebox.showerror("Error",                         "Something went wrong..!!!"                         "\nPlease Try again")     pass</pre> 
To calculate cumulative CGPA of both semesters	<pre>else:     self.cgpa = (float(self.firsttgpavalue['text']) + float(self.secondtgpavalue['text']))/2     self.cgpa = "{:.2f}".format(self.cgpa)     self.cgpavalue['text'] = str(self.cgpa)</pre>	<ol style="list-style-type: none"> <li>If user didn't enter all values-</li> </ol> <pre>messagebox.showwarning("Warning..!!",                       "Provide all required details"                       "\nthen Proceed..")</pre> <ol style="list-style-type: none"> <li>Exception handling</li> </ol> <pre>except:     messagebox.showerror("Error",                         "Something went wrong..!!!"                         "\nPlease Try again")     pass</pre>
To see the grade point relation	<pre>def give_grade_info(self):     self.impp = tkinter.Toplevel()     self.impp.title('Grade - credit')     self.impp.minsize(399, 631)     self.impp.maxsize(399, 631)     self.canvas = tkinter.Canvas(self.impp, width=399, height=631)     self.canvas.grid()     self.photu = tkinter.PhotoImage(file='GradePoint.png', master=root)     self.canvas.create_image(0, 0, image=self.photu, anchor=NW)     self.impp.mainloop()     pass</pre>	<p>Toplevel() function is used to create new window in which grade and credits are given, for this we have used an image to reduce the coding part and make it attractive. The image is inserted inside the canvas &amp; then packed with grid.</p>

To clear all the entered and calculated values	<pre>def clear_text_command(self, ):     self.firstsub1.delete(0, END), self.firstsub2.delete(0, END), self.firstsub3.delete(0, END)     self.firstsub4.delete(0, END), self.firstcourse1.delete(0, END), self.firstcourse2.delete(0, END)     self.firstcourse3.delete(0, END), self.firstcourse4.delete(0, END), self.firstcredit1.delete(0, END)     self.firstcredit2.delete(0, END), self.firstcredit3.delete(0, END), self.firstcredit4.delete(0, END)     self.firstgrade1.delete(0, END), self.firstgrade2.delete(0, END), self.firstgrade3.delete(0, END)     self.firstgrade4.delete(0, END), self.secondsub1.delete(0, END), self.secondsub2.delete(0, END)     self.secondsub3.delete(0, END), self.secondsub4.delete(0, END)     self.secondcourse1.delete(0, END), self.secondcourse2.delete(0, END), self.secondcourse3.delete(0, END)     self.secondcourse4.delete(0, END), self.secondgrade1.delete(0, END), self.secondgrade2.delete(0, END)     self.secondgrade3.delete(0, END), self.secondgrade4.delete(0, END), self.secondcredit1.delete(0, END)     self.secondcredit2.delete(0, END), self.secondcredit3.delete(0, END), self.secondcredit4.delete(0, END)     self.firsttgpavalue.config(text=0.00), self.secondstgpavalue.config(text=0.00), self.cgpa.value.config(text=0.00)</pre>	This function clears all the inserted and calculated values, this is done with the help of – <ol style="list-style-type: none"> <li>1. Delete function for the entered values.</li> <li>2. Config(text=0.00) function for calculated values</li> </ol>
--	---	--

## FLOW CHART



# GUI Screenshots



HOME PAGE

Grade point vs Grade

Grade	Grade point
O	10
A+	9
A	8
B+	7
B	6
C+	5
C	4
E (reapper)	0
F (fail)	0
G (backlog)	0
I (result incomplete)	0

CGPA Calculator

# CGPA CALCULATOR

**First Semester :**

Course Title	Course Code	Credit	Grade Obtained
CPP Programming	CSE202	3	O
C Programming	CSE101	3	A+
Python Programming	INT213	3	O
Data Structure	CSE205	4	O

**Calculate TGPA**      **9.77**

**Please find your CGPA and take steps for improvements**

**Second Semester :**

Course Title	Course Code	Credit	Grade Obtained
Integration Maths	MTH166	3	A+
Computer Networks	CSE306	3	A+
Discrete Mathematics	MTH401	3	O
DBMS	INT306	4	D

**Calculate TGPA**      **9.54**

**Your CGPA is :** **9.65**

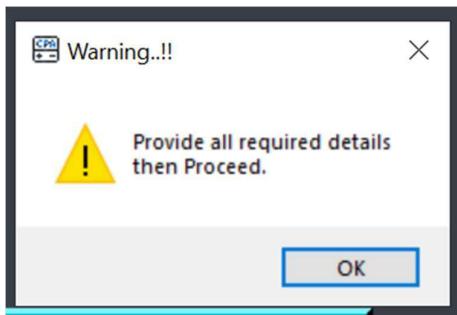
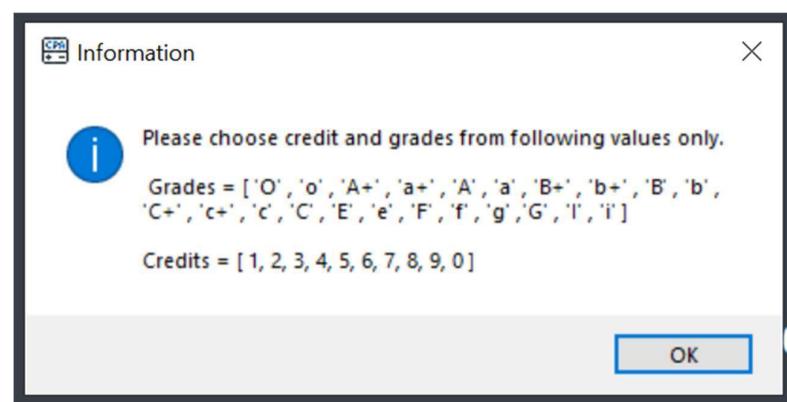
**Grade Info**

**CLEAR ALL**

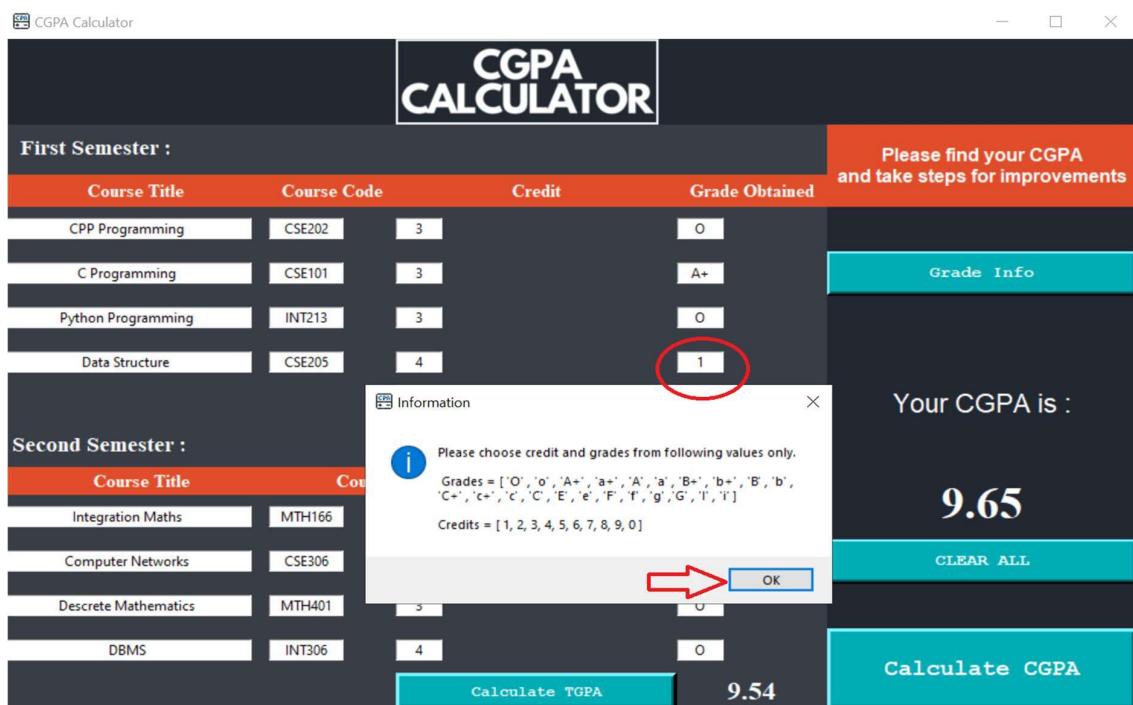
**Calculate CGPA**

### Result as TGPA's and CGPA

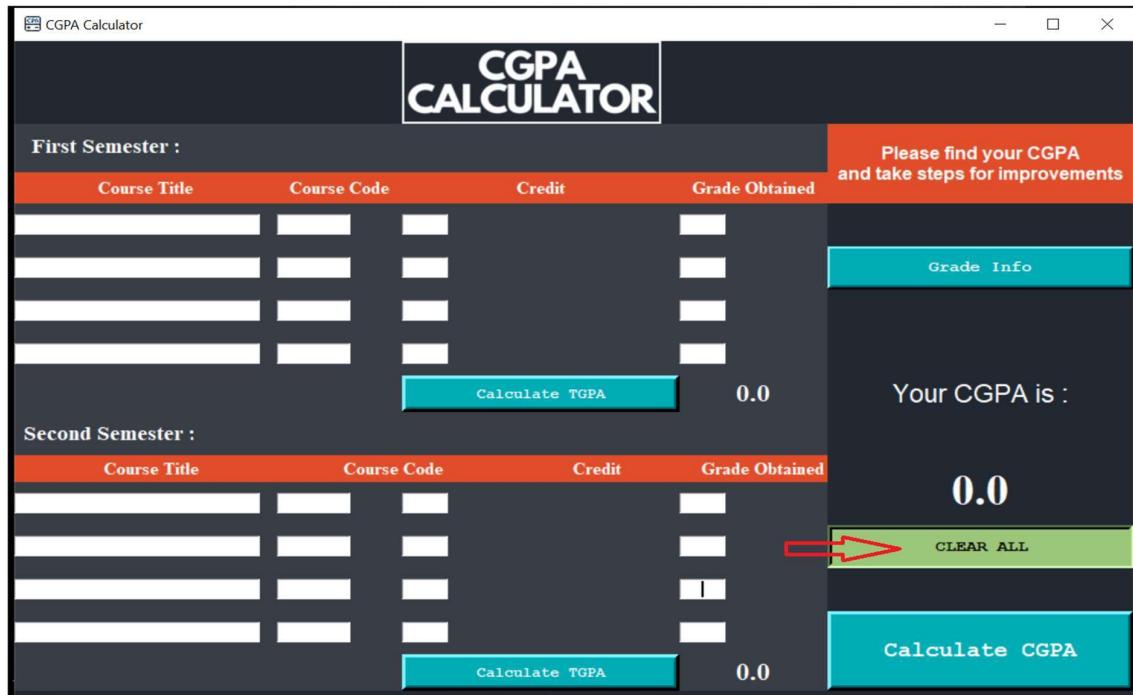
**Will show Warning as well as Info if we put Wrong Grade or Credit**



**Warning of incorrect input**



Above is the example of incorrect input



Clear All option to delete all inputs and reset the result in one press

# Source Code

## 1. Calculator.py - file

```
import tkinter
from tkinter import *
from tkinter import messagebox

grade1 = []
grade1point = []
credit1 = []
grade2 = []
grade2point = []
credit2 = []

class Main(object):
    _images = []

    def __init__(self, root):
        self._frame_p = tkinter.Frame(root, background="#222831", )
        self._images.append(tkinter.PhotoImage('CGPALOGO.png',
file='CGPALOGO.png'))
        self._label_1 = tkinter.Label(image="CGPALOGO.png", )
        self._label_1.grid(in_=self._frame_p, column=1, row=1,
columnspan=1, ipadx=0, ipady=0, padx=0, pady=0,
                    rowspan=2,
                    sticky="nsew")
        self._frame_1 = tkinter.Frame(root, background="#393E46", )
        self._frame_2 = tkinter.Frame(root, background="#393E46", )
        self._frame_3 = tkinter.Frame(root, background="#393E46", )
        self._frame_4 = tkinter.Frame(root, background="#393E46", )
        self._frame_5 = tkinter.Frame(root, background="#393E46", )
        self._frame_6 = tkinter.Frame(root, background="#393E46", )
        self._frame_7 = tkinter.Frame(root, background="#393E46", )
        self._frame_8 = tkinter.Frame(root, background="#393E46", )
        self._frame_9 = tkinter.Frame(root, background="#393E46", )
        self._frame_10 = tkinter.Frame(root, background="#393E46", )
        self._frame_11 = tkinter.Frame(root, background="#393E46", )
        self._frame_12 = tkinter.Frame(root, background="#393E46", )
        self._frame_13 = tkinter.Frame(root, background="#393E46", )
        self._frame_14 = tkinter.Frame(root, background="#393E46", )
        self._frame_15 = tkinter.Frame(root, background="#393E46", )
        self._frame_16 = tkinter.Frame(root, background="#393E46", )
        self._frame_17 = tkinter.Frame(root, background="#393E46", )
        self._frame_18 = tkinter.Frame(root, background="#393E46", )
        self._frame_19 = tkinter.Frame(root, background="#393E46", )
        self._frame_20 = tkinter.Frame(root, background="#393E46", )
        self._frame_21 = tkinter.Frame(root, background="#393E46", )
        self._frame_24 = tkinter.Frame(root, background="#393E46", )
        self._frame_22 = tkinter.Frame(root, background="#393E46", )
        self._frame_23 = tkinter.Frame(root, background="#393E46", )
        self._frame_25 = tkinter.Frame(root, background="#393E46", )
        self._frame_26 = tkinter.Frame(root, background="#393E46", )
        self._frame_27 = tkinter.Frame(root, background="#393E46", )
        self._frame_28 = tkinter.Frame(root, background="#393E46", )
        self._frame_29 = tkinter.Frame(root, background="#393E46", )
        self._frame_30 = tkinter.Frame(root, background="#393E46", )
        self._frame_32 = tkinter.Frame(root, background="#393E46", )
```

```

        self._frame_33 = tkinter.Frame(root, background="#393E46", )
        self._frame_34 = tkinter.Frame(root, background="#393E46", )
        self._frame_35 = tkinter.Frame(root, background="#393E46", )
        self._frame_42 = tkinter.Frame(root, background="#393E46", )
        self._frame_43 = tkinter.Frame(root, background="#393E46", )
        self._frame_44 = tkinter.Frame(root, background="#393E46", )
        self._frame_45 = tkinter.Frame(root, background="#393E46", )
        self._frame_46 = tkinter.Frame(root, background="#393E46", )
        self._label_12 = tkinter.Label(self._frame_1,
                                     activebackground="#393E46", activeforeground="#1A374D",
                                     background="#393E46", font="{Times New Roman} 14 bold", foreground="#FFFFFF",
                                     text=" First Semester :", padx=8,
                                     pady=8)
        self._label_13 = tkinter.Label(root, activebackground="#E14D2A",
                                     background="#E14D2A",
                                     font="{Times New Roman} 12 bold",
                                     text="Course Title", foreground="#FFFFFF")
        self._label_14 = tkinter.Label(root, activebackground="#E14D2A",
                                     background="#E14D2A",
                                     font="{Times New Roman} 12 bold",
                                     text="Course Code", foreground="#FFFFFF")
        self._label_16 = tkinter.Label(root, activebackground="#E14D2A",
                                     background="#E14D2A", foreground="#FFFFFF",
                                     font="{Times New Roman} 12 bold",
                                     text="Credit", )
        self._label_20 = tkinter.Label(root, activebackground="#E14D2A",
                                     background="#E14D2A", foreground="#FFFFFF",
                                     font="{Times New Roman} 12 bold",
                                     text="Grade Obtained", )
        self.firstsub1 = tkinter.Entry(self._frame_2, justify="center",
                                      width=35, )
        self.firstcourse1 = tkinter.Entry(self._frame_3, justify="center",
                                         width=10, )
        self.firstcredit1 = tkinter.Entry(self._frame_4, justify="center",
                                         width=0, )
        self.firstgrade1 = tkinter.Entry(self._frame_5, justify="center",
                                         width=0, )
        self.firstsub2 = tkinter.Entry(self._frame_6, justify="center",
                                      width=35, )
        self.firstcourse2 = tkinter.Entry(self._frame_7, justify="center",
                                         width=10, )
        self.firstcredit2 = tkinter.Entry(self._frame_8, justify="center",
                                         width=0, )
        self.firstgrade2 = tkinter.Entry(self._frame_9, justify="center",
                                         width=0, )
        self.firstsub3 = tkinter.Entry(self._frame_10, justify="center",
                                      width=35, )
        self.firstcourse3 = tkinter.Entry(self._frame_11, justify="center",
                                         width=10, )
        self.firstcredit3 = tkinter.Entry(self._frame_12, justify="center",
                                         width=0, )
        self.firstgrade3 = tkinter.Entry(self._frame_13, justify="center",
                                         width=0, )
        self.firstsub4 = tkinter.Entry(self._frame_14, justify="center",
                                      width=35, )
        self.firstcourse4 = tkinter.Entry(self._frame_15, justify="center",
                                         width=10, )
        self.firstcredit4 = tkinter.Entry(self._frame_16, justify="center",
                                         width=0, )
        self.firstgrade4 = tkinter.Entry(self._frame_17, justify="center",
                                         width=0, )

```

```

width=0, )
    self.clear_all = tkinter.Button(root, activebackground="#9cc67a",
background="#00ADB5", borderwidth=4,
                                         cursor="hand2", font="{Courier New} 11
bold", foreground="#FFFFFF", text="CLEAR ALL",)

    self.gradeInfo = tkinter.Button(root, activebackground="#9cc67a",
background="#00ADB5", borderwidth=4,
                                         cursor="hand2", font="{Courier New} 11
bold", foreground="#FFFFFF", text="Grade Info",)
        self.firsttgpa = tkinter.Button(root, activebackground="#9cc67a",
background="#00ADB5", borderwidth=5,
                                         cursor="hand2", font="{Courier New}
10 bold", foreground="#FFFFFF",
                                         text="Calculate TGPA", )

    self.firsttgpavalue = tkinter.Label(root,
activebackground="#393E46", activeforeground="#1A374D",
                                         foreground="#FFFFFF",
                                         background="#393E46",
font="{Times New Roman} 17 bold",
                                         text=0.00, )

    self._label_39 = tkinter.Label(self._frame_20,
activebackground="#393E46", activeforeground="#1A374D",
                                         background="#393E46",
foreground="#FFFFFF", font="{Times New Roman} 14 bold",
                                         text=" Second Semester : ", )

    self._label_11 = tkinter.Label(self._frame_24,
activebackground="#E14D2A", background="#E14D2A",
                                         foreground="#FFFFFF",
                                         font="{Times New Roman} 12 bold",
text="Course Title", )

    self._label_15 = tkinter.Label(self._frame_24,
activebackground="#E14D2A", background="#E14D2A",
                                         foreground="#FFFFFF",
                                         font="{Times New Roman} 12 bold",
text="Course Code", )

    self._label_17 = tkinter.Label(self._frame_24,
activebackground="#E14D2A", background="#E14D2A",
                                         foreground="#FFFFFF",
                                         font="{Times New Roman} 12 bold",
text="Credit", )

    self._label_18 = tkinter.Label(self._frame_24,
activebackground="#E14D2A", background="#E14D2A",
                                         foreground="#FFFFFF",
                                         font="{Times New Roman} 12 bold",
text="Grade Obtained", )

    self.secondtgpa = tkinter.Button(root, activebackground="#9cc67a",
background="#00ADB5", borderwidth=5,
                                         cursor="hand2", font="{Courier
New} 10 bold", foreground="#FFFFFF",
                                         text="Calculate TGPA", )

    self.secondtgpavalue = tkinter.Label(root,
activebackground="#393E46", activeforeground="#1A374D",
                                         background="#393E46",
font="{Times New Roman} 17 bold",
                                         foreground="#FFFFFF",
                                         text=0.00, )

    self.secondsub1 = tkinter.Entry(self._frame_22, justify="center",
width=35, )
        self.secondcourse1 = tkinter.Entry(self._frame_23,
justify="center", width=10, )

```

```

        self.secondcredit1 = tkinter.Entry(self._frame_25,
justify="center", width=0, )
        self.secondgradel = tkinter.Entry(self._frame_26, justify="center",
width=0, )
        self.secondsub2 = tkinter.Entry(self._frame_27, justify="center",
width=35, )
        self.secondcourse2 = tkinter.Entry(self._frame_28,
justify="center", width=10, )
        self.secondcredit2 = tkinter.Entry(self._frame_29,
justify="center", width=0, )
        self.secondgrade2 = tkinter.Entry(self._frame_30, justify="center",
width=0)
        self.secondsub3 = tkinter.Entry(self._frame_34, justify="center",
width=35, )
        self.secondcourse3 = tkinter.Entry(self._frame_33,
justify="center", width=10, )
        self.secondcredit3 = tkinter.Entry(self._frame_32,
justify="center", width=0, )
        self.secondgrade3 = tkinter.Entry(self._frame_42, justify="center",
width=0, )
        self.secondsub4 = tkinter.Entry(self._frame_43, justify="center",
width=35, )
        self.secondcourse4 = tkinter.Entry(self._frame_44,
justify="center", width=10, )
        self.secondcredit4 = tkinter.Entry(self._frame_45,
justify="center", width=0, )
        self.secondgrade4 = tkinter.Entry(self._frame_46, justify="center",
width=0, )
        self.cgpa = tkinter.Button(root, activebackground="#9cc67a",
background="#00ADB5", borderwidth=5,
                                         cursor="hand2", font="{Courier New} 15
bold", foreground="#FFFFFF",
                                         text="Calculate CGPA", )
        self._label_7 = tkinter.Label(root, borderwidth=8,
activebackground="#E14D2A", background="#E14D2A",
                                         font="{Wide Latin} 12 bold",
foreground="#FFFFFF",
                                         text="""Please find your CGPA
and take steps for improvements""", pady=10)
        self._label_32 = tkinter.Label(root, borderwidth=8,
activebackground="#222831", background="#222831",
                                         font="{Wide Latin} 17",
foreground="#FFFFFF",
                                         text="""Your CGPA is :""", )
        self.cgpavalue = tkinter.Label(root, activebackground="#222831",
activeforeground="#1A374D",
                                         background="#222831", font="{Times
New Roman} 30 bold", foreground="#FFFFFF",
                                         text=0.00, )

# widget commands
self.clear_all.configure(command=self.clear_text_command)
self.gradeInfo.configure(command=self.give_grade_info)
self.firsttgpa.configure(command=self.firsttgpa_command)
self.secondtgpa.configure(command=self.secondtgpa_command)
self.cgpa.configure(command=self.cgpa_command)

# geometry
self._frame_p.grid(column=3, row=0, sticky="news")
self._frame_1.grid(in_=root, column=1, row=4, columnspan=4,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,

```

```

                sticky="news")
        self._frame_2.grid(in_=root, column=1, row=6, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_3.grid(in_=root, column=2, row=6, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_4.grid(in_=root, column=3, row=6, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_5.grid(in_=root, column=4, row=6, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_6.grid(in_=root, column=1, row=7, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_7.grid(in_=root, column=2, row=7, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_8.grid(in_=root, column=3, row=7, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_9.grid(in_=root, column=4, row=7, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_10.grid(in_=root, column=1, row=8, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_11.grid(in_=root, column=2, row=8, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_12.grid(in_=root, column=3, row=8, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_13.grid(in_=root, column=4, row=8, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_14.grid(in_=root, column=1, row=9, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_15.grid(in_=root, column=2, row=9, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_16.grid(in_=root, column=3, row=9, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_17.grid(in_=root, column=4, row=9, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_18.grid(in_=root, column=1, row=10, columnspan=2,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_20.grid(in_=root, column=1, row=11, columnspan=4,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_24.grid(in_=root, column=1, row=12, columnspan=4,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_22.grid(in_=root, column=1, row=13, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")

```

```

        self._frame_23.grid(in_=root, column=2, row=13, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_25.grid(in_=root, column=3, row=13, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_26.grid(in_=root, column=4, row=13, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_27.grid(in_=root, column=1, row=14, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_28.grid(in_=root, column=2, row=14, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_29.grid(in_=root, column=3, row=14, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_30.grid(in_=root, column=4, row=14, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_32.grid(in_=root, column=3, row=15, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_33.grid(in_=root, column=2, row=15, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_34.grid(in_=root, column=1, row=15, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_35.grid(in_=root, column=1, row=17, columnspan=2,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_42.grid(in_=root, column=4, row=15, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_43.grid(in_=root, column=1, row=16, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_44.grid(in_=root, column=2, row=16, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_45.grid(in_=root, column=3, row=16, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._frame_46.grid(in_=root, column=4, row=16, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                      sticky="news")
        self._label_7.grid(in_=root, column=5, row=4, columnspan=2,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=2,
                      sticky="nsew")
        self._label_11.grid(in_=self._frame_24, column=1, row=1,
columnspan=1, ipadx=0, ipady=0, padx=0, pady=0,
                      rowspan=1, sticky="nsew")
        self._label_15.grid(in_=self._frame_24, column=2, row=1,
columnspan=1, ipadx=0, ipady=0, padx=0, pady=0,
                      rowspan=1, sticky="nsew")
        self._label_17.grid(in_=self._frame_24, column=3, row=1,
columnspan=1, ipadx=0, ipady=0, padx=0, pady=0,
                      rowspan=1, sticky="nsew")
        self._label_18.grid(in_=self._frame_24, column=4, row=1,

```

```

    colspan=1, ipadx=0, ipady=0, padx=0, pady=0,
                rowspan=1, sticky="nsew")
        self._label_12.grid(in_=self._frame_1, column=1, row=1,
    colspan=1, ipadx=0, ipady=0, padx=0, pady=0,
                rowspan=1, sticky="w")
        self._label_13.grid(in_=root, column=1, row=5, colspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                sticky="nsew")
        self._label_14.grid(in_=root, column=2, row=5, colspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                sticky="nsew")
        self._label_16.grid(in_=root, column=3, row=5, colspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                sticky="nsew")
        self._label_20.grid(in_=root, column=4, row=5, colspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                sticky="nsew")
        self._label_32.grid(in_=root, column=5, row=9, colspan=2,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=3,
                sticky="nsew")
        self._label_39.grid(in_=self._frame_20, column=1, row=1,
colspan=1, ipadx=0, ipady=0, padx=0,
                rowspan=1, sticky="w")
        self.firstsub1.grid(in_=self._frame_2, column=1, row=1,
colspan=1, ipadx=0, ipady=0, padx=0,
                rowspan=1, sticky="ew")
        self.firstcourse1.grid(in_=self._frame_3, column=1, row=1,
colspan=1, ipadx=0, ipady=0, padx=0,
                rowspan=1, sticky="ew")
        self.firstcredit1.grid(in_=self._frame_4, column=1, row=1,
colspan=1, ipadx=0, ipady=0, padx=0,
                rowspan=5, sticky="ew", )
        self.firstgrade1.grid(in_=self._frame_5, column=1, row=1,
colspan=1, ipadx=0, ipady=0, padx=0,
                rowspan=1, sticky="ew")
        self.firstsub2.grid(in_=self._frame_6, column=1, row=1,
colspan=1, ipadx=0, ipady=0, padx=0,
                rowspan=1, sticky="ew")
        self.firstcourse2.grid(in_=self._frame_7, column=1, row=1,
colspan=1, ipadx=0, ipady=0, padx=0,
                rowspan=1, sticky="ew")
        self.firstcredit2.grid(in_=self._frame_8, column=1, row=1,
colspan=1, ipadx=0, ipady=0, padx=0,
                rowspan=1, sticky="ew")
        self.firstgrade2.grid(in_=self._frame_9, column=1, row=1,
colspan=1, ipadx=0, ipady=0, padx=0,
                rowspan=1, sticky="ew")
        self.firstsub3.grid(in_=self._frame_10, column=1, row=1,
colspan=1, ipadx=0, ipady=0, padx=0,
                rowspan=1, sticky="ew")
        self.firstcourse3.grid(in_=self._frame_11, column=1, row=1,
colspan=1, ipadx=0, ipady=0, padx=0,
                rowspan=1, sticky="ew")
        self.firstcredit3.grid(in_=self._frame_12, column=1, row=1,
colspan=1, ipadx=0, ipady=0, padx=0,
                rowspan=1, sticky="ew")
        self.firstgrade3.grid(in_=self._frame_13, column=1, row=1,
colspan=1, ipadx=0, ipady=0, padx=0,
                rowspan=1, sticky="ew")
        self.firstsub4.grid(in_=self._frame_14, column=1, row=1,
colspan=1, ipadx=0, ipady=0, padx=0,

```

```

        rowspan=1, sticky="ew")
    self.firstcourse4.grid(in_=self._frame_15, column=1, row=1,
columnspan=1, ipadx=0, ipady=0, padx=0, pady=0,
                           rowspan=1, sticky="ew")
    self.firstcredit4.grid(in_=self._frame_16, column=1, row=1,
columnspan=1, ipadx=0, ipady=0, padx=0, pady=0,
                           rowspan=1, sticky="ew")
    self.firstgrade4.grid(in_=self._frame_17, column=1, row=1,
columnspan=1, ipadx=0, ipady=0, padx=0, pady=0,
                           rowspan=1, sticky="ew")
    self.firsttgpa.grid(in_=root, column=3, row=10, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                           sticky="nsew")
    self.firsttgpavalue.grid(in_=root, column=4, row=10, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                           sticky="nsew")
    self.secondtgpa.grid(in_=root, column=3, row=17, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                           sticky="nsew")
    self.secondtgpavalue.grid(in_=root, column=4, row=17, columnspan=1,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                           sticky="nsew")
    self.secondsub1.grid(in_=self._frame_22, column=1, row=1,
columnspan=1, ipadx=0, ipady=0, padx=0, pady=0,
                           rowspan=1, sticky="ew")
    self.secondcourse1.grid(in_=self._frame_23, column=1, row=1,
columnspan=1, ipadx=0, ipady=0, padx=0, pady=0,
                           rowspan=1, sticky="ew")
    self.secondcredit1.grid(in_=self._frame_25, column=1, row=1,
columnspan=1, ipadx=0, ipady=0, padx=0, pady=0,
                           rowspan=1, sticky="ew")
    self.secondgrade1.grid(in_=self._frame_26, column=1, row=1,
columnspan=1, ipadx=0, ipady=0, padx=0, pady=0,
                           rowspan=1, sticky="ew")
    self.secondsub2.grid(in_=self._frame_27, column=1, row=1,
columnspan=1, ipadx=0, ipady=0, padx=0, pady=0,
                           rowspan=1, sticky="ew")
    self.secondcourse2.grid(in_=self._frame_28, column=1, row=1,
columnspan=1, ipadx=0, ipady=0, padx=0, pady=0,
                           rowspan=1, sticky="ew")
    self.secondcredit2.grid(in_=self._frame_29, column=1, row=1,
columnspan=1, ipadx=0, ipady=0, padx=0, pady=0,
                           rowspan=1, sticky="ew")
    self.secondgrade2.grid(in_=self._frame_30, column=1, row=1,
columnspan=1, ipadx=0, ipady=0, padx=0, pady=0,
                           rowspan=1, sticky="ew")
    self.secondsub3.grid(in_=self._frame_34, column=1, row=1,
columnspan=1, ipadx=0, ipady=0, padx=0, pady=0,
                           rowspan=1, sticky="ew")
    self.secondcourse3.grid(in_=self._frame_33, column=1, row=1,
columnspan=1, ipadx=0, ipady=0, padx=0, pady=0,
                           rowspan=1, sticky="ew")
    self.secondcredit3.grid(in_=self._frame_32, column=1, row=1,
columnspan=1, ipadx=0, ipady=0, padx=0, pady=0,
                           rowspan=1, sticky="ew")
    self.secondgrade3.grid(in_=self._frame_42, column=1, row=1,
columnspan=1, ipadx=0, ipady=0, padx=0, pady=0,
                           rowspan=1, sticky="ew")
    self.secondsub4.grid(in_=self._frame_43, column=1, row=1,
columnspan=1, ipadx=0, ipady=0, padx=0, pady=0,
                           rowspan=1, sticky="ew")

```

```

        self.secondcourse4.grid(in_=self._frame_44, column=1, row=1,
columnspan=1, ipadx=0, ipady=0, padx=0, pady=0,
                           rowspan=1, sticky="ew")
    self.secondcredit4.grid(in_=self._frame_45, column=1, row=1,
columnspan=1, ipadx=0, ipady=0, padx=0, pady=0,
                           rowspan=1, sticky="ew")
    self.secondgrade4.grid(in_=self._frame_46, column=1, row=1,
columnspan=1, ipadx=0, ipady=0, padx=0, pady=0,
                           rowspan=1, sticky="ew")
    self.clear_all.grid(in_=root, column=5, row=14, columnspan=3,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                           sticky="nsew")
    self.gradeInfo.grid(in_=root, column=5, row=7, columnspan=3,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=1,
                           sticky="nsew")
    self.cgpa.grid(in_=root, column=5, row=16, columnspan=3, ipadx=0,
ipady=0, padx=0, pady=0, rowspan=2,
                           sticky="nsew")
    self.cgpavalue.grid(in_=root, column=5, row=12, columnspan=2,
ipadx=0, ipady=0, padx=0, pady=0, rowspan=2,
                           sticky="nsew")

    self._frame_10.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_10.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_24.grid_rowconfigure(1, weight=0, minsize=5, pad=0)
    self._frame_24.grid_columnconfigure(1, weight=0, minsize=240,
pad=0)
    self._frame_24.grid_columnconfigure(2, weight=0, minsize=181,
pad=0)
    self._frame_24.grid_columnconfigure(3, weight=0, minsize=175,
pad=0)
    self._frame_24.grid_columnconfigure(4, weight=1, minsize=67, pad=0)
    self._frame_25.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_25.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_26.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_26.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_27.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_27.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_28.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_28.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_29.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_29.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_3.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_3.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_30.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_30.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_32.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_32.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_33.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_33.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_34.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_34.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_4.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_4.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_42.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_42.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_43.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_43.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_44.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_44.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
    self._frame_45.grid_rowconfigure(1, weight=0, minsize=40, pad=0)

```

```

self._frame_45.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
self._frame_46.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
self._frame_46.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
self._frame_5.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
self._frame_5.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
self._frame_6.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
self._frame_6.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
self._frame_7.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
self._frame_7.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
self._frame_8.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
self._frame_8.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
self._frame_9.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
self._frame_9.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
self._frame_10.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
self._frame_10.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
self._frame_11.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
self._frame_11.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
self._frame_12.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
self._frame_12.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
self._frame_13.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
self._frame_13.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
self._frame_14.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
self._frame_14.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
self._frame_15.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
self._frame_15.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
self._frame_16.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
self._frame_16.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
self._frame_17.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
self._frame_17.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
self._frame_2.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
self._frame_2.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
self._frame_20.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
self._frame_20.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
self._frame_22.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
self._frame_22.grid_columnconfigure(1, weight=0, minsize=40, pad=0)
self._frame_23.grid_rowconfigure(1, weight=0, minsize=40, pad=0)
self._frame_23.grid_columnconfigure(1, weight=0, minsize=40, pad=0)

class CustomMain(Main):
    print("CGPA Calculator Opened")

    def clear_text_command(self, ):
        self.firstsub1.delete(0, END)
        self.firstsub2.delete(0, END)
        self.firstsub3.delete(0, END)
        self.firstsub4.delete(0, END)
        self.firstcourse1.delete(0, END)
        self.firstcourse2.delete(0, END)
        self.firstcourse3.delete(0, END)
        self.firstcourse4.delete(0, END)
        self.firstcredit1.delete(0, END)
        self.firstcredit2.delete(0, END)
        self.firstcredit3.delete(0, END)
        self.firstcredit4.delete(0, END)
        self.firstgrade1.delete(0, END)
        self.firstgrade2.delete(0, END)
        self.firstgrade3.delete(0, END)
        self.firstgrade4.delete(0, END)
        self.secondsub1.delete(0, END)
        self.secondsub2.delete(0, END)

```

```

        self.secondsub3.delete(0, END)
        self.secondsub4.delete(0, END)
        self.secondcourse1.delete(0, END)
        self.secondcourse2.delete(0, END)
        self.secondcourse3.delete(0, END)
        self.secondcourse4.delete(0, END)
        self.secondgrade1.delete(0, END)
        self.secondgrade2.delete(0, END)
        self.secondgrade3.delete(0, END)
        self.secondgrade4.delete(0, END)
        self.secondcredit1.delete(0, END)
        self.secondcredit2.delete(0, END)
        self.secondcredit3.delete(0, END)
        self.secondcredit4.delete(0, END)
        self.firsttgpavalue.config(text=0.00)
        self.secondtgpa.value.config(text=0.00)
        self.cgpa.value.config(text=0.00)

    def give_grade_info(self):
        self.impp = tkinter.Toplevel()
        self.impp.title('Grade - credit')
        self.impp.minsize(399, 631)
        self.impp.maxsize(399, 631)
        self.canvas = tkinter.Canvas(self.impp, width=399, height=631)
        self.canvas.grid()
        self.photu = tkinter.PhotoImage(file='GradePoint.png', master=root)
        self.canvas.create_image(0, 0, image=self.photu, anchor=NW)
        self.impp.mainloop()
        pass

    def firsttgpa_command(self, ):
        try:
            print("Semester1 TGPA Button Pressed.")
            list = ['O', 'o', 'A+', 'a+', 'A', 'a', 'B+', 'b+', 'B', 'b',
            'C+', 'c+', 'c', 'C', 'E', 'e', 'F', 'f', 'g',
            'G', 'I', 'i']
            num = [1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
            if (self.firstgrade1.get() in list) and (self.firstgrade2.get() in list) and (
                self.firstgrade3.get() in list) and \
                (self.firstgrade4.get() in list) and
                (int(self.firstcredit1.get()) in num) and (
                    int(self.firstcredit2.get()) in num) and \
                    (int(self.firstcredit3.get()) in num) and
                    (int(self.firstcredit4.get()) in num):

                a1, b1, c1, d1 = self.firstcredit1.get(),
                self.firstcredit2.get(), self.firstcredit3.get(), self.firstcredit4.get()
                credit1.append(int(a1))
                credit1.append(int(b1))
                credit1.append(int(c1))
                credit1.append(int(d1))
                print(credit1)
                e1, f1, g1, h1 = self.firstgrade1.get(),
                self.firstgrade2.get(), self.firstgrade3.get(), self.firstgrade4.get()
                e1 = e1.upper()
                grade1.append(e1)
                f1 = f1.upper()
                grade1.append(f1)
                g1 = g1.upper()

```

```

grade1.append(g1)
h1 = h1.upper()
grade1.append(h1)
print(grade1)
y = 0
while y < 4:
    if grade1[y] == "O":
        number = 10
        gradelpoint.append(number)
    elif grade1[y] == "A+":
        number = 9
        gradelpoint.append(number)
    elif grade1[y] == "A":
        number = 8
        gradelpoint.append(number)
    elif grade1[y] == "B+":
        number = 7
        gradelpoint.append(number)
    elif grade1[y] == "B":
        number = 6
        gradelpoint.append(number)
    elif grade1[y] == "C+":
        number = 5
        gradelpoint.append(number)
    elif grade1[y] == "C":
        number = 4
        gradelpoint.append(number)
    elif grade1[y] == "D+":
        number = 3
        gradelpoint.append(number)
    elif grade1[y] == "D":
        number = 2
        gradelpoint.append(number)
    elif grade1[y] == "E":
        number = 0
        gradelpoint.append(number)
    y = y + 1
print(gradelpoint)
i = 0
self.tgpal, first_part1, second_part1 = 0.00, 0, 0
while i < 4:
    first_part1 += credit1[i] * gradelpoint[i]
    second_part1 += credit1[i]
    i = i + 1
self.tgpal = first_part1 / second_part1
self.tgpal = "{:.2f}".format(self.tgpal)
print(self.tgpal)
self.firsttgpavalue["text"] = self.tgpal
else:
    messagebox.showinfo("Information",
                        "Please choose credit and grades from
following values only.\n\n Grades = [ 'O' , 'o' , 'A+' , 'a+' , 'A' , 'a' ,
'B+' , 'b+' , 'B' , 'b' , 'C+' , 'c+' , 'C' , 'c' , 'E' , 'e' , 'F' , 'f' ,
'g' , 'G' , 'I' , 'i' ]\n\nCredits = [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 ]")
except:
    messagebox.showerror("Error", "Something went
wrong..!!!\nPlease Try again")
    pass

def secondtgpa_command(self):
    try:

```

```

        print("Semester2 TGPA Button Pressed.")
        list = ['O', 'o', 'A+', 'a+', 'A', 'a', 'B+', 'b+', 'B', 'b',
'C+', 'c+', 'c', 'C', 'E', 'e', 'F', 'f', 'g',
'G', 'I', 'i']
        num = [1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
        if (self.firstgrade1.get() in list) and (self.firstgrade2.get()
in list) and (
            self.firstgrade3.get() in list) and \
            (self.firstgrade4.get() in list) and
(int(self.firstcredit1.get()) in num) and (
            int(self.firstcredit2.get()) in num) and \
            (int(self.firstcredit3.get()) in num) and
(int(self.firstcredit4.get()) in num):
            a2, b2, c2, d2 = self.secondcredit1.get(),
self.secondcredit2.get(), self.secondcredit3.get(),
self.secondcredit4.get()
            credit2.append(int(a2))
            credit2.append(int(b2))
            credit2.append(int(c2))
            credit2.append(int(d2))
            print(credit2)
            e2, f2, g2, h2 = self.secondgrade1.get(),
self.secondgrade2.get(), self.secondgrade3.get(), self.secondgrade4.get()
            e2 = e2.upper()
            grade2.append(e2)
            f2 = f2.upper()
            grade2.append(f2)
            g2 = g2.upper()
            grade2.append(g2)
            h2 = h2.upper()
            grade2.append(h2)
            print(grade2)
            z = 0
            while z < 4:
                if grade2[z] == "O":
                    number = 10
                    grade2point.append(number)
                elif grade2[z] == "A+":
                    number = 9
                    grade2point.append(number)
                elif grade2[z] == "A":
                    number = 8
                    grade2point.append(number)
                elif grade2[z] == "B+":
                    number = 7
                    grade2point.append(number)
                elif grade2[z] == "B":
                    number = 6
                    grade2point.append(number)
                elif grade2[z] == "C+":
                    number = 5
                    grade2point.append(number)
                elif grade2[z] == "C":
                    number = 4
                    grade2point.append(number)
                elif grade2[z] == "D+":
                    number = 3
                    grade2point.append(number)
                elif grade2[z] == "D":
                    number = 2
                    grade2point.append(number)

```

```

        elif grade2[z] == "E":
            number = 0
            grade2point.append(number)
            z = z + 1
    print(grade2point)
j = 0
self.tgpa2, first_part2, second_part2 = 0.00, 0, 0
while j < 4:
    first_part2 += credit2[j] * grade2point[j]
    second_part2 += credit2[j]
    j = j + 1
self.tgpa2 = first_part2 / second_part2
self.tgpa2 = "{:.2f}".format(self.tgpa2)
print(self.tgpa2)
self.secondtgpavalue["text"] = self.tgpa2
print("tgpa1=", self.tgpa1)
print("tgpa2=", self.tgpa2)
else:
    messagebox.showinfo("Information",
                        "Please choose credit and grades from
following values only.\n\n Grades = [ 'O', 'o', 'A+', 'a+', 'A', 'a', 'B+', 'B',
'b+', 'B', 'b', 'C+', 'c+', 'C', 'C', 'E', 'e', 'F', 'f', 'g', 'G', 'I',
'i']\nCredits= [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 ]")
except:
    messagebox.showerror("Error", "Something went
wrong..!!!\nPlease Try again")
pass

def cgpa_command(self, ):
    try:
        print("CGPA Button Pressed.")
        if self.firstcredit1.get() == "" and self.firstcredit2.get() ==
"" and self.firstcredit3.get() == "" and \
           self.firstcredit4.get() == "" and
self.secondcredit1.get() == "" and self.secondcredit2.get() == "" and
self.secondcredit3.get() == "" and \
           self.secondcredit4.get() == "" and
self.firstgrade1.get() == "" and self.firstgrade2.get() == "" and
self.firstgrade3.get() == "" and \
           self.firstgrade4.get() == "" and
self.secondgrade1.get() == "" and self.secondgrade2.get() == "" and
self.secondgrade3.get() == "" and \
           self.secondgrade4.get() == "":
            messagebox.showwarning("Warning..!!", "Provide all required
details\nthen Proceed.")
        else:
            self.cgp = (float(self.firstattgpavalue['text']) +
float(self.secondtgpavalue['text'])) / 2
            self.cgp = "{:.2f}".format(self.cgp)
            self.cgpavalue['text'] = str(self.cgp)

    except:
        messagebox.showerror("Error", "Something went
wrong..!!!\nPlease Try again")
    pass

if __name__ == '__main__':
    root = Tk()
    demo = CustomMain(root)
    root.title('CGPA Calculator')

```

```

root.maxsize(978, 610)
root.minsize(976, 609)
root.wm_iconbitmap('Icon.ico')
root.configure(bg="#222831")
root.mainloop()

```

## 2. Logic\_builder.py - code

```

grades1 = []
grade1point=[]
credit1 = []
grades2 = []
grade2point=[]
credit2 = []

#Collects the data of Class names and Grades in Letter Form
def collect():
    print("Enter 8 credits of first semester")
    y,x,a,z=0,0,0,0
    while (y <8):
        credit= int(input(""))
        credit1.append(credit)
        y = y +1
    print(credit1)
    print("Enter 8 grades of first semester")
    while(z<8):
        grades=input("")
        grades=grades.upper()
        if(grades=="O"):
            number=10
            grade1point.append(number)
        elif(grades=="A+"):
            number=9
            grade1point.append(number)
        elif(grades=="A"):
            number=8
            grade1point.append(number)
        elif(grades=="B+"):
            number=7
            grade1point.append(number)
        elif(grades=="B"):
            number=6
            grade1point.append(number)
        elif(grades=="C+"):
            number=5
            grade1point.append(number)
        elif(grades=="C"):
            number=4
            grade1point.append(number)
        elif(grades=="D+"):
            number=3
            grade1point.append(number)
        elif(grades=="D"):
            number=2
            grade1point.append(number)
        elif(grades=="E"):
            number=1
            grade1point.append(number)

```

```

        number=0
        grade1point.append(number)

        grades1.append(grades)
        z = z + 1
print(grade1point)
print(grades1)
print("Enter 9 credits of second semester")
while (x <9):
    credit= int(input(""))
    credit2.append(credit)
    x = x +1
print(credit2)
print("Enter 9 grades of second semester")
while(a<9):
    grades=input("")
    grades=grades.upper()
    if(grades=="O"):
        number=10
        grade2point.append(number)
    elif(grades=="A+"):
        number=9
        grade2point.append(number)
    elif(grades=="A"):
        number=8
        grade2point.append(number)
    elif(grades=="B+"):
        number=7
        grade2point.append(number)
    elif(grades=="B"):
        number=6
        grade2point.append(number)
    elif(grades=="C+"):
        number=5
        grade2point.append(number)
    elif(grades=="C"):
        number=4
        grade2point.append(number)
    elif(grades=="D+"):
        number=3
        grade2point.append(number)
    elif(grades=="D"):
        number=2
        grade2point.append(number)
    elif(grades=="E"):
        number=0
        grade2point.append(number)
    grades2.append(grades)
    a= a + 1
print(grade2point)
print(grades2)
calculate()

def calculate():
    i,j=0,0
    tgpa1,tgpa2,first_part1,second_part1,first_part2,second_part2=0,0,0,0
    ,0,0
    while(i<8):
        first_part1+=credit1[i]*gradepoint[i]
        second_part1+=credit1[i]

```

```
i=i+1
while(j<9):
    first_part2+=credit2[j]*grade2point[j]
    second_part2+=credit2[j]
    j=j+1
tgpal=first_part1/second_part1
tgpa2=first_part2/second_part2
tgpal=float("{:.2f}".format(tgpal))
tgpa2=float("{:.2f}".format(tgpa2))
print("Tgpal=",tgpal)
print("Tgpa2=",tgpa2)
cgpa=(tgpal+tgpa2)/2
print(cgpa)
print("Cgpa=",cgpa)
pass
collect()
```

## Conclusion

Through this project the students would be able to easily calculate their TGPA's and CGPA's in a faster and easier manner and as developers we were also able to learn the features of python programming language and understand the working of the graphical user interface how to use the tkinter library.

We were also able to come up with certain new ideas which we have implemented in our project. By making this project we have explored the vast field of python programming language and were able to use our theoretical knowledge in a practical project which has helped us in understanding the concepts far better than any bookish knowledge.

# References

## Books:

- Introduction to Programming Using Python - Y. Liang
- Head-First Python, 2nd edition
- Python Cookbook

## Websites:

- [https://www.tutorialspoint.com/python/python\\_gui\\_programming.htm](https://www.tutorialspoint.com/python/python_gui_programming.htm)
- <https://stackoverflow.com>
- <https://www.geeksforgeeks.org/pythonprogramming-language>
- [https://www.tutorialspoint.com/python/python\\_gui\\_programming.htm](https://www.tutorialspoint.com/python/python_gui_programming.htm)
- <http://www.w3resource.com/python/python-tutorial.php>

THANK YOU