SGPA CALCULATOR

Personal Mini Project Using Tkinter

Kavya Reddy Vutukuri
SRM INSTITUTE OF SCIENCE & TECHNOLOGY ,CHENNAI

Table of Contents

INTRODUCTION

This project aims to develop a Graphical User Interface (GUI) based SGPA calculator using Python programming language and Tkinter library. The application takes inputs such as name, registration number, semester, department, and subject-wise grades from the user and calculates the Semester Grade Point Average (SGPA) based on the grading system of the university.

The project uses a MySQL database to store the information entered by the user. The data is stored in a well-organized manner that facilitates easy retrieval and manipulation of data. The database can store a large amount of data, which makes the application suitable for use in universities and colleges.

The GUI design of the application is intuitive and user-friendly, which makes it easy for users to interact with the application. The application also provides features such as adding, editing, and deleting student records, and generating reports based on the stored data.

Overall, this project provides a simple and effective solution to calculate SGPA for university students. It also demonstrates the use of programming languages and databases to develop real-world applications.

DESIGN

Algorithm

1. Start the program.
2. Import the necessary libraries: mysql.connector, tkinter, and math.
3. Connect to the MySQL database using mysql.connector.
4. Create a cursor object to execute SQL queries.
5. Create a tkinter window with labels and entry boxes for the user to input their registration number, semester, and department.
6. Create a button for the user to click to calculate their SGPA.
7. When the user clicks the button, retrieve the input values from the entry boxes.
8. Construct an SQL query to retrieve the relevant information for the input registration number, semester, and department.
9. Execute the SQL query using the cursor object.
10. Retrieve the result set from the SQL query and store it in a variable.
11. If the result set is empty, display an error message to the user.
12. If the result set is not empty, iterate over the rows in the result set.
13. For each row, calculate the total credit hours and total grade points earned by multiplying the credit hours for each course by the grade point earned for that course and adding the results together.
14. Calculate the SGPA by dividing the total grade points earned by the total credit hours.
15. Round the SGPA to two decimal places using the math library.
16. Display the calculated SGPA to the user in a new tkinter window.
17. Close the database connection.
18. End the program.

Pseudocode:

```
program sgpa_calculator;
uses
  Tkinter, Math;
var
  root, label1, label2, label3, label4: Widget;
  entry1, entry2, entry3, entry4: Entry;
  button1: Button;

function calculate_sgpa(): Real;
var
  total_credits, credits_obtained, grade_points: Integer;
  grade: String;
  i: Integer;
begin
  total_credits := 0;
  credits_obtained := 0;
  for i := 1 to 4 do
  begin
    grade := entry3[i].get();
    if grade = 'O' then
      grade_points := 10
    else if grade = 'A+' then
      grade_points := 9
    else if grade = 'A' then
      grade_points := 8
    else if grade = 'B+' then
      grade_points := 7
    else if grade = 'B' then
      grade_points := 6
    else if grade = 'C' then
      grade_points := 5
    else if grade = 'F' then
      grade_points := 0
    else if grade = '' then
      grade_points := 0;
    total_credits := total_credits + StrToInt(Abentry2[i].get());
    credits_obtained := credits_obtained + (grade_points *
StrToInt(entry2[i].get()));
  end;
  calculate_sgpa := credits_obtained / total_credits;
end;

procedure display_sgpa();
begin
  label4.configure('text', 'Your SGPA is: ' +
FloatToStrF(calculate_sgpa(), ffFixed, 6, 2));
end;

begin
  root := Tk();
  root.title('SGPA Calculator');

  label1 := Label(root, 'text', 'Enter your name:');
  label1.pack();
  entry1 := Entry(root);
  entry1.pack();

  label2 := Label(root, 'text', 'Enter your roll number:');
  label2.pack();
  entry2 := Entry(root);
  entry2.pack();

  label3 := Label(root, 'text', 'Enter your grades (O, A+, A, B+,
B, C, P, F):');
  label3.pack();
  entry3[1] := Entry(root);
  entry3[1].pack();
  entry3[2] := Entry(root);
  entry3[2].pack();
  entry3[3] := Entry(root);
  entry3[3].pack();
  entry3[4] := Entry(root);
  entry3[4].pack();

  label4 := Label(root, 'text', '');
  label4.pack();

  button1 := Button(root, 'text', 'Calculate SGPA', command :=
display_sgpa);
  button1.pack();

  root.mainloop();
end.
```

Development

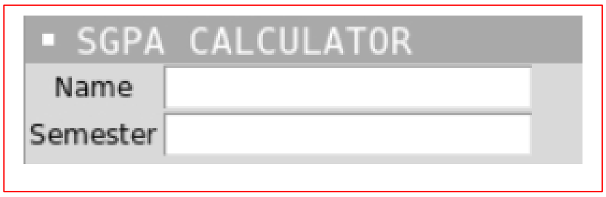
Source Code

```
# Import tkinter as tk

import tkinter as tk


# creating a new tkinter window

window = tk.Tk()


# assigning a title

window.title("SGPA CALCULATOR")


# specifying geometry for window size

window.geometry("700x270")
```

Tkinter module is imported, window is created. Geometry of the window is set

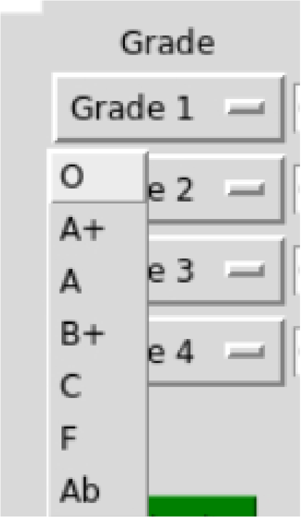
```
# declaring objects for entering data

Entry1 = tk.Entry(window)

Entry2 = tk.Entry(window)

Entry3 = tk.Entry(window)

#Entry4 = tk.Entry(window)

#Entry5 = tk.Entry(window)

#Entry6 = tk.Entry(window)

#Entry7 = tk.Entry(window)

Entry8 = tk.Entry(window)

Entry9 = tk.Entry(window)

Entry10 = tk.Entry(window)

Entry11 = tk.Entry(window)

Entry12 = tk.Entry(window)

Entry13 = tk.Entry(window)
```

All the entry widgets defined for entering data



```
Entry14 = tk.Entry(window)
Entry15 = tk.Entry(window)

tot=0
total=0
GradeL=['O','A+','A','B+','C','F','Ab']



Entry4=tk.StringVar()
Entry4.set("Grade 1")
dropG1=tk.OptionMenu(window,Entry4,*GradeL)
dropG1.grid(row=5,column=2)

Entry5=tk.StringVar()
Entry5.set("Grade 2")
dropG1=tk.OptionMenu(window,Entry5,*GradeL)
dropG1.grid(row=6,column=2)

Entry6=tk.StringVar()
Entry6.set("Grade 3")
dropG1=tk.OptionMenu(window,Entry6,*GradeL)
dropG1.grid(row=7,column=2)

Entry7=tk.StringVar()
Entry7.set("Grade 4")
dropG1=tk.OptionMenu(window,Entry7,*GradeL)
dropG1.grid(row=8,column=2)
```

Grades for the drop down list



```
# function to display the total credits and SGPA
```

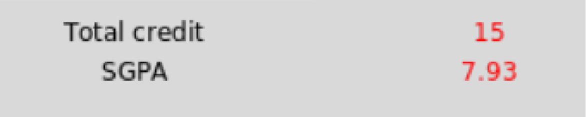
```
def display():

    # Variable to store total marks
    global tot
    global total
    tot = 0


    # give total credits for grade A
    if Entry4.get() == "O":
        tot += (int(cv1.get())*10)

    # total credits for grade B
    if Entry4.get() == "A+":
        tot += (int(cv1.get())*9)

    # total credits for grade C
    if Entry4.get() == "A":
        tot += (int(cv1.get())*8)

     # give total credits for grade D
    if Entry4.get() == "B+":
        tot += (int(cv1.get())*7)


    # total credits for grade P
    if Entry4.get() == "B":

        tot += (int(cv1.get())*6)

    if Entry4.get() == "C":
```

```
    tot += (int(cv1.get())*5)

# total credits for grade F
if Entry4.get() == "F":

    tot += 0

if Entry4.get() == "Ab":
    tot += 0

# 10*number of subject credits
# give total credits for grade A
if Entry5.get() == "O":
    tot += (int(cv2.get())*10)

# 9*number of subject credits give
# total credits for grade B
if Entry5.get() == "A+":

    tot += (int(cv2.get())*9)

# 8*number of subject credits give
# total credits for grade C
if Entry5.get() == "A":

    tot += (int(cv2.get())*8)

# 7*number of subject credits
# give total credits for grade D
if Entry5.get() == "B+":
    tot += (int(cv2.get())*7)

 # 6*number of subject credits give
 # total credits for grade P
 if Entry5.get() == "B":

    tot += (int(cv2.get())*6)

 if Entry5.get() == "C":

    tot += (int(cv2.get())*5)

 # 0*number of subject credits give
 # total credits for grade F
 if Entry5.get() == "F":

    tot += 0

 if Entry5.get() == "Ab":

    tot += 0

# 10*number of subject credits
 # give total credits for grade A
 if Entry6.get() == "O":

    # grid method is used for placing
    # the widgets at respective positions
    # in table like structure .

    tot += (int(cv3.get())*10)
```

```
# 9*number of subject credits give
# total credits for grade B
if Entry6.get() == "A+":

    tot += (int(cv3.get())*9)

# 8*number of subject credits give
# total credits for grade C
if Entry6.get() == "A":
   tot += (int(cv3.get())*8)

# 7*number of subject credits
# give total credits for grade D
if Entry6.get() == "B+":
    tot += (int(cv3.get())*7)

# 6*number of subject credits give
# total credits for grade P
if Entry6.get() == "B":

    tot += (int(cv3.get())*6)

if Entry6.get() == "C":

    tot += (int(cv3.get())*5)

# 0*number of subject credits give
# total credits for grade F
if Entry6.get() == "F":
    tot += 0

  if Entry6.get() == "Ab":

    tot += 0



 # 10*number of subject credits
  # give total credits for grade A
  if Entry7.get() == "O":

    # grid method is used for placing
    # the widgets at respective positions
    # in table like structure .

    tot += (int(cv4.get())*10)

  # 9*number of subject credits give
  # total credits for grade B
  if Entry7.get() == "A+":

    tot += (int(cv4.get())*9)

  # 8*number of subject credits give
  # total credits for grade C
  if Entry7.get() == "A":

    tot += (int(cv4.get())*8)

  # 7*number of subject credits
  # give total credits for grade D
```

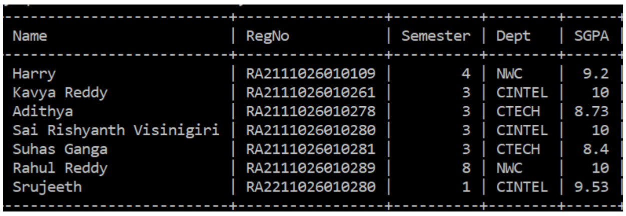
```
if Entry7.get() == "B+":

    tot += (int(cv4.get())*7)

# 6*number of subject credits give
# total credits for grade P
if Entry7.get() == "B":

    tot += (int(cv4.get())*6)

if Entry7.get() == "C":

    tot += (int(cv4.get())*5)

# 0*number of subject credits give
# total credits for grade F
if Entry7.get() == "F":

    tot += 0

if Entry7.get() == "Ab":

    tot += 0
# to display total credits
Tot=int(cv1.get())+int(cv2.get())+int(cv3.get())+int(cv4.get())
tk.Label(window, text=str(Tot),fg='red').grid(row=9, column=4)


# to display SGPA
total=round(tot/Tot,2)
tk.Label(window, text=str(total),fg='red').grid(row=1
```

Displays the total credits obtained and SGPA



```
Total credit     15
    SGPA       7.93
    import mysql.connector as sqltor   # Imports database programming package

    mycon=sqltor.connect(host="localhost",user="root",passwd="abcde",database="GPA")

    # Establishes connecton with the database dropdead stored in mysql

    mycursor=mycon.cursor()   #  Creates a cursor instance that is used to execute queries on the
database


    query="insert into Student(Name,RegNo,Semester,Dept,SGPA)
values('{}','{}',{},'{}',{})".format(Entry1.get(),Entry2.get(),Entry3.get(),Entry15.get(),total)

    # Inserts the Winner's name,score in the table Leaderboard

    mycursor.execute(query)  # Query executed

    mycon.commit()

    # Permanent changes made to the table

    mycon.close()   # Cleans up the environment
```

Inserts the student details in database



```
----------------------------+------------------+------------+--------+------
 Name                       | RegNo            | Semester | Dept   | SGPA |
----------------------------+------------------+------------+--------+------
 Harry                      | RA2111026010109 |        4 | NWC    |  9.2 |
 Kavya Reddy                | RA2111026010261 |        3 | CINTEL |   10 |
 Adithya                    | RA2111026010278 |        3 | CTECH  | 8.73 |
 Sai Rishyanth Visinigiri   | RA2111026010280 |        3 | CINTEL |   10 |
 Suhas Ganga                | RA2111026010281 |        3 | CTECH  |  8.4 |
 Rahul Reddy                | RA2111026010289 |        8 | NWC    |   10 |
 Srujeeth                   | RA2211026010280 |        1 | CINTEL | 9.53 |
----------------------------+------------------+------------+--------+------
```

```
# end of display function


# label to enter name

tk.Label(window, text="Name").grid(row=0, column=0)


# label for registration number

tk.Label(window, text="Reg.No").grid(row=0, column=3)


# label for roll Number

tk.Label(window, text="Semester").grid(row=1, column=0)


# label for dept.

#tk.Label(window,text="Dept.").grid(row=1,coloumn=3)



# labels for serial numbers
```

```
tk.Label(window, text="Srl.No").grid(row=4, column=0)

tk.Label(window, text="1").grid(row=5, column=0)

tk.Label(window, text="2").grid(row=6, column=0)

tk.Label(window, text="3").grid(row=7, column=0)

tk.Label(window, text="4").grid(row=8, column=0)


tk.Label(window, text="Subject").grid(row=4, column=1)


sub=['18CSS207J','18CSC205J','18CSS202J','18MAB202T','18CSC208L','18PDH103T','18CSC204J','18CY
M101T','18MAB102T']


curVar=tk.StringVar()

curVar.set("Subject 1")

drop=tk.OptionMenu(window,curVar,*sub)

drop.grid(row=5,column=1)



curVar2=tk.StringVar()

curVar2.set("Subject 2")

drop2=tk.OptionMenu(window,curVar2,*sub)

drop2.grid(row=6,column=1)


curVar3=tk.StringVar()

curVar3.set("Subject 3")


drop3=tk.OptionMenu(window,curVar3,*sub)

drop3.grid(row=7,column=1)


curVar4=tk.StringVar()

curVar4.set("Subject 4")

drop4=tk.OptionMenu(window,curVar4,*sub)
```

```
drop4.grid(row=8,column=1)

#Entry11.grid(row=5, column=1)
#Entry12.grid(row=6, column=1)
#Entry13.grid(row=7, column=1)
#Entry14.grid(row=8, column=1)

# tk.Label(window, text="18CSC203J").grid(row=3, column=1)
# tk.Label(window, text="18PDT102T").grid(row=4, column=1)
# tk.Label(window, text="18MAB204T").grid(row=5, column=1)
# tk.Label(window, text="18CSS201J").grid(row=6, column=1)

# label for grades
tk.Label(window, text="Grade").grid(row=4, column=2)
#Entry4.grid(row=5, column=2)
#Entry5.grid(row=6, column=2)
#Entry6.grid(row=7, column=2)
#Entry7.grid(row=8, column=2)

# labels for subject credits
cv1 = tk.StringVar(value=0)
cv2 = tk.StringVar(value=0)
cv3 = tk.StringVar(value=0)
cv4 = tk.StringVar(value=0)
#Spinbox for subcredits
tk.Label(window, text="Sub Credit").grid(row=4, column=3)
w1=tk.Spinbox(window,from_=0, to_=5,textvariable=cv1,wrap=True).grid(row=5,column=3)
w2=tk.Spinbox(window,from_=0, to_=5,textvariable=cv2,wrap=True).grid(row=6,column=3)
w3=tk.Spinbox(window,from_=0, to_=5,textvariable=cv3,wrap=True).grid(row=7,column=3)
w4=tk.Spinbox(window,from_=0, to_=5,textvariable=cv4,wrap=True).grid(row=8,column=3)
```

```
'''tk.Label(window, text="4").grid(row=3, column=3)
tk.Label(window, text="4").grid(row=4, column=3)
tk.Label(window, text="3").grid(row=5, column=3)
tk.Label(window, text="4").grid(row=6, column=3)'''

# taking entries of name, reg, roll number respectively
Entry1 = tk.Entry(window)
Entry2 = tk.Entry(window)
Entry3 = tk.Entry(window)

# organizing them in the grid
Entry1.grid(row=0, column=1)
Entry2.grid(row=0, column=4)
Entry3.grid(row=1, column=1)
Entry15.grid(row=1,column=4)
tk.Label(window,text="Dept.").grid(row=1,column=3)
#Entry15.grid(row=1,column=4)

# button to display all the calculated credit scores and sgpa
button1 = tk.Button(window, text="submit", bg="green", command=display)
button1.grid(row=11, column=2)
 tk.Label(window, text="Total credit").grid(row=9, column=3)
tk.Label(window, text="SGPA").grid(row=10, column=3)

window.mainloop()
```

OUTPUT - SCREENSHOTS

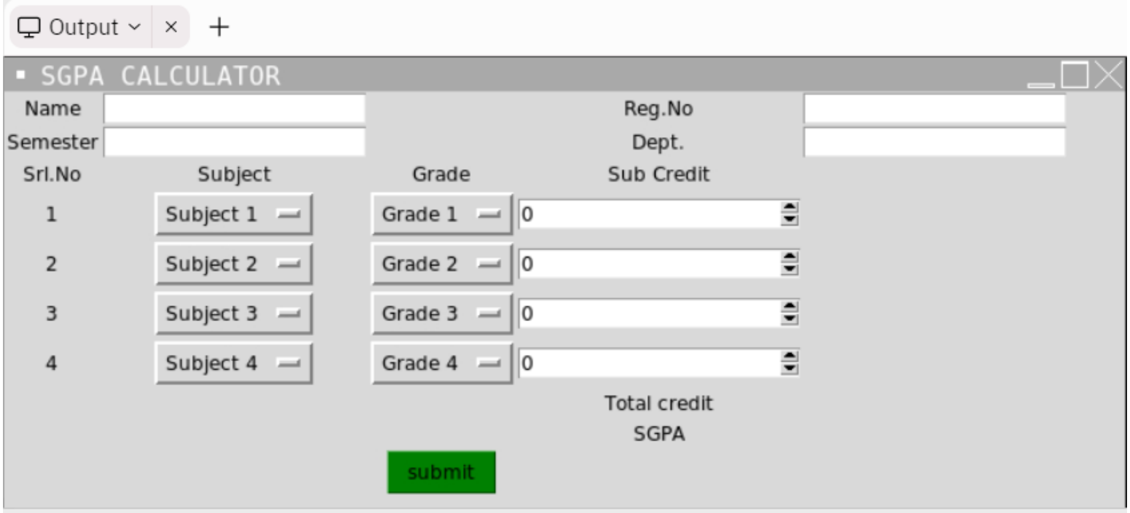
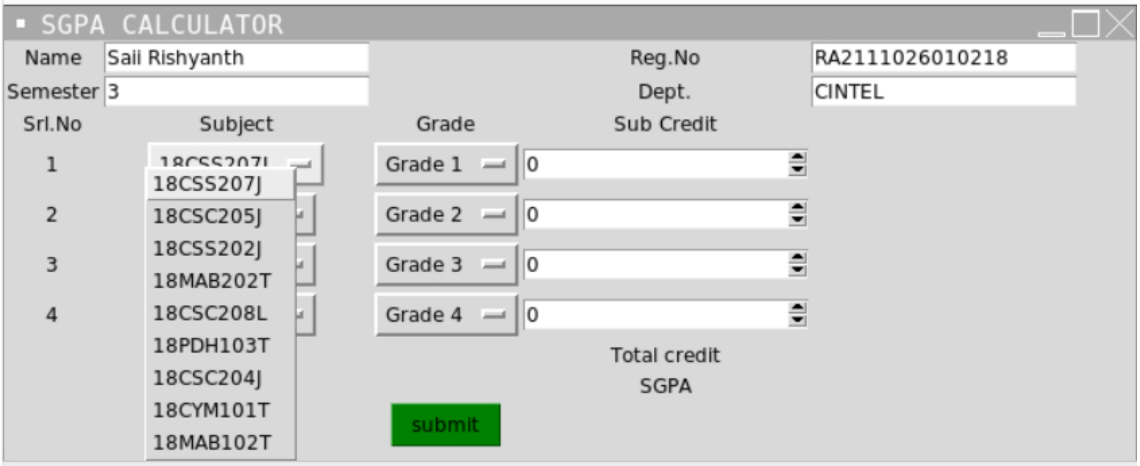


Figure 1: SGPA Calculator Window

The SGPA calculator window contains entry fields to enter data such as name, reg no, current semester and department. There are no of widgets such combobox to select among different subjects and grades. The spinbox widgets allows to select the sub credit for each subject. The submit button when clicked will calculate the total credit obtained and sgpa of the student.



Name Saii Rishyanth

Semester 3

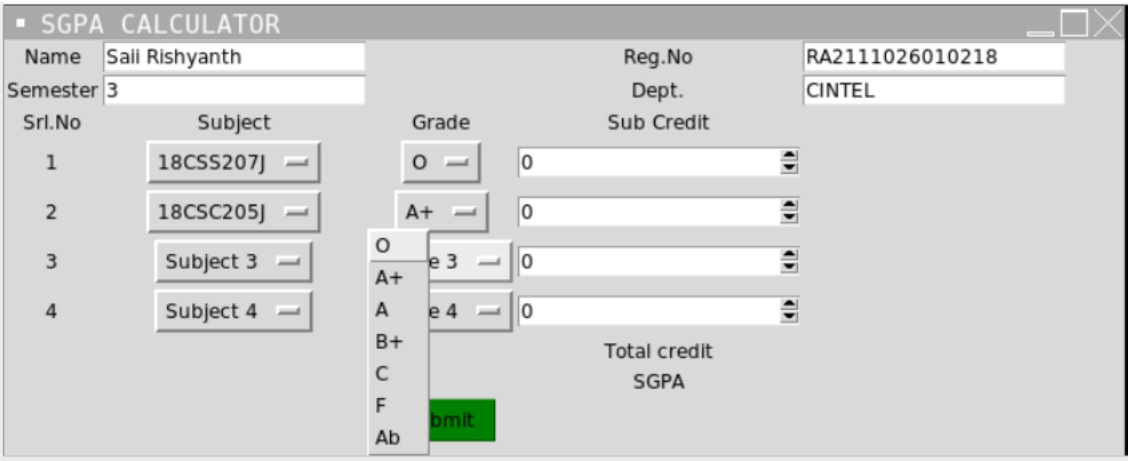
Reg.No RA2111026010218

Dept. CINTEL

18CSS207J
18CSC205J
18CSS202J
18MAB202T
18CSC208L
18PDH103T
18CSC204J
18CYM101T
18MAB102T

Figure 2: Combobox for Subjects

The combobox for subjects allows users to select among various subjects

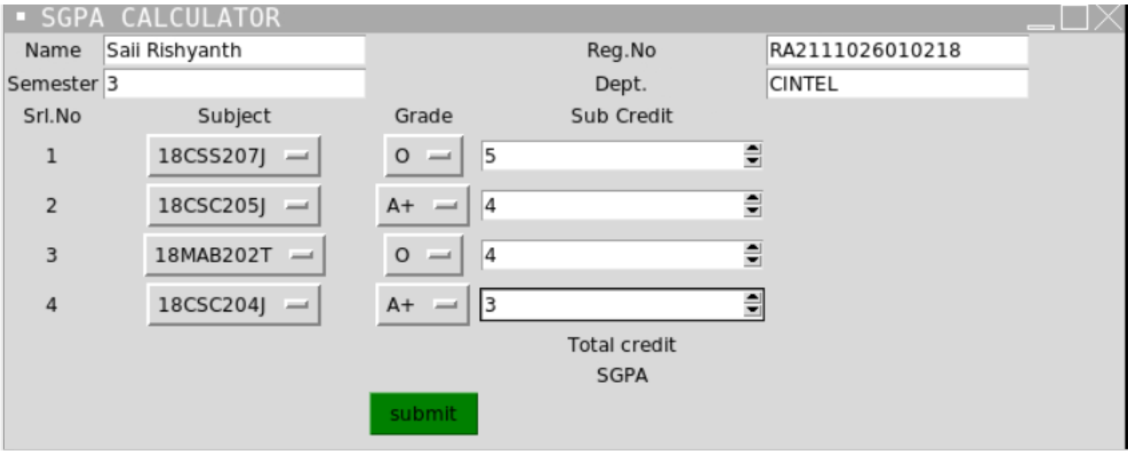


Name	Saii Rishyanth	Reg.No	RA2111026010218
Semester	3	Dept.	CINTEL

Srl.No	Subject	Grade	Sub Credit
1	18CSS207J	O	0
2	18CSC205J	A+	0
3	Subject 3	e 3	0
4	Subject 4	e 4	0

Figure 3: Combobox for Grades

The combobox for grades allows users to select the grade he achieved for a particular subject

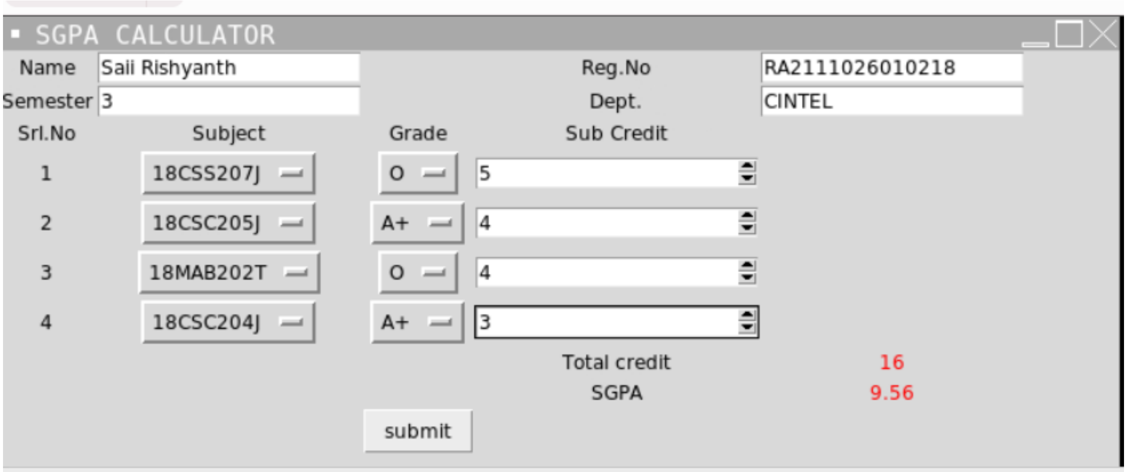


Name	Saii Rishyanth	Reg.No	RA2111026010218
Semester	3	Dept.	CINTEL

Srl.No	Subject	Grade	Sub Credit
1	18CSS207J	O	5
2	18CSC205J	A+	4
3	18MAB202T	O	4
4	18CSC204J	A+	3

Figure 4: Spinbox widget for sub-credits

The spinbox has a range between 0 to 5 to select the sub credit of the subject



Name	Saii Rishyanth	Reg.No	RA2111026010218
Semester	3	Dept.	CINTEL

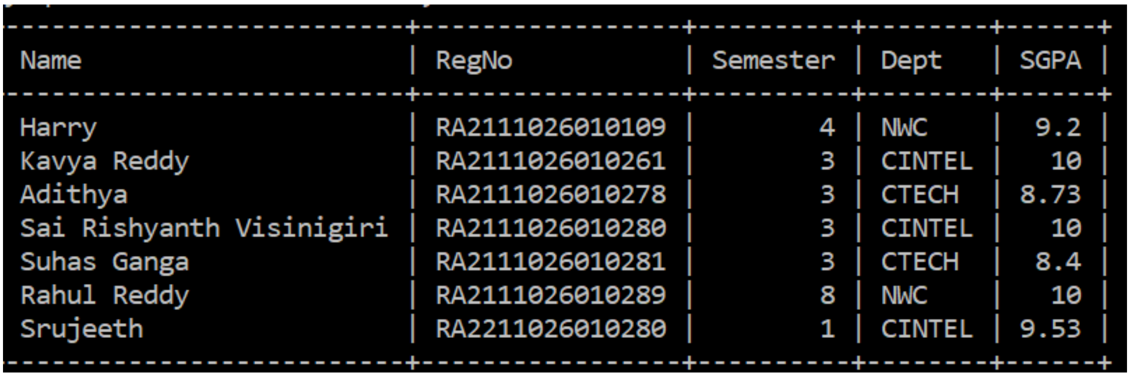
Srl.No	Subject	Grade	Sub Credit
1	18CSS207J	O	5
2	18CSC205J	A+	4
3	18MAB202T	O	4
4	18CSC204J	A+	3

Total credit 16

SGPA 9.56

Figure 5: Display of Total Credit & SGPA after clicking Submit button

The display function is called when the Submit button is clicked which performs the sgpa calculation and displays it to the user.



```
-------------------------+----------------+----------+--------+------+
 Name                    | RegNo          | Semester | Dept   | SGPA |
-------------------------+----------------+----------+--------+------+
 Harry                   | RA2111026010109 |        4 | NWC    |  9.2 |
 Kavya Reddy             | RA2111026010261 |        3 | CINTEL |   10 |
 Adithya                 | RA2111026010278 |        3 | CTECH  | 8.73 |
 Sai Rishyanth Visinigiri | RA2111026010280 |        3 | CINTEL |   10 |
 Suhas Ganga             | RA2111026010281 |        3 | CTECH  |  8.4 |
 Rahul Reddy             | RA2111026010289 |        8 | NWC    |   10 |
 Srujeeth                | RA2211026010280 |        1 | CINTEL | 9.53 |
-------------------------+----------------+----------+--------+------+
```

Figure 6: Database containing Student GPA details

The database GPA stores the student name, registration number, semester, department and SGPA details.

Future Scope

→ Integration with more universities: The current implementation of the SGPA calculator is designed for a specific university. In the future, the application can be extended to work with more universities, allowing students from different institutions to calculate their SGPA.

→ Mobile Application: The application can be developed into a mobile application that can be installed on smartphones. This will make it more convenient for students to access their SGPA information on-the-go.

→ Integration with Online Learning Management Systems: The SGPA calculator can be integrated with online learning management systems, such as Moodle or Blackboard. This will allow students to access their SGPA information from within their course management systems.

→ Statistical Analysis: The SGPA calculator can be extended to provide statistical analysis of the student's academic performance. The system can generate graphs and charts to show the student's performance over time.

→ Machine Learning: Machine learning algorithms can be used to predict a student's SGPA based on their past academic performance. This will allow students to plan their academic goals and achieve better results.

→ Social Media Integration: The SGPA calculator can be integrated with social media platforms to allow students to share their SGPA information with their friends and family. This will encourage healthy competition among students and motivate them to perform better.

→ Cloud-based Infrastructure: The SGPA calculator can be deployed on cloud-based infrastructure, making it more scalable and flexible. This will allow for more users to access the application simultaneously, without compromising on performance.

Conclusion

In conclusion, the SGPA calculator project is a useful tool for students to calculate their semester grades and GPA. It is a simple yet effective solution that can save time and effort for students who want to focus on their studies instead of manually calculating their grades. The project is also scalable and can be extended to include additional features such as course recommendations based on performance and integration with other academic systems. By using modern technologies such as Python, Tkinter, and MySQL, this project provides a practical example of how computer science can be applied to solve real-world problems in education.

Bibliography

https://www.tutorialspoint.com/python/python_gui_programming.htm#:~:text=Tkinter%20is%20the%20standard%20GUI,Tkinter%20is%20an%20easy%20task.

https://docs.python.org/3/library/tkinter.html

https://www.geeksforgeeks.org/introduction-to-tkinter/