Investigatory Project

Computer Science

SCHOOL MANAGEMENT SYSTEM (SMS)

Done By

ARUNACHALAM S R

KISHORE KUMAR R

CERTIFICATE

This is to certify that ________________________ of class XII of Krishnaswamy Vidyanikethan, Cuddalore has completed their project file under my guidance. They have taken proper care and shown utmost sincerity in completing this project.

I certify that this project is up to my expectations and as per guidelines issued by CBSE

Signature of Teacher

Signature of Internal Examiner Signature of External Examiner

Signature of Principal

ACKNOWLEDGEMENT

I wish to express my deep gratitude and sincere thanks to all my teachers for encouragement and the management for providing all facilities to successfully complete the project work.

I extend my sincere thanks to my Principal, **Mr. K.P.Jills** and my Computer Science teacher, **Mrs. K.Aranganayagi** whose valuable guidance helped me not only successfully complete the project but also appreciate the beauty of the subject and python language.

I extend my gratitude to my parents and classmates for their valuable support and time.

INDEX

S.NO	CONTENT	PAGE
1	Certificate	2
2	Acknowledgement	3
3	About Project	5
4	System Software and Hardware	6
5	Python	7
6	Language Features	8-9
7	Designing and Coding	10-19
8	Code	20-242
9	Output	243-254
10	Future Enhancement	255
11	Bibliography	256

ABOUT PROJECT

In a school, it is very essential to manage the records of the teachers and students. To do so, we hereby developed this SCHOOL MANAGEMENT SYSTEM. In doing so we reduce the workload and time consumed to search the details of the teachers and students etc...

Without proper school management, a school may run in lack of efficiency. This system can be used to store the details and to retrieve and display it when necessary. The program will have a login for admin, teacher, and office and allow them to add, modify, view and delete the details of the students and teachers.

By developing this project in python, we understand the usage of user-defined modules and by connecting it with SQL we learn to manipulate data that is stored at the backend.

In conclusion, we can say, this project focuses on school management using simple data stored in tables and making use of user-defined modules.

SYSTEM SOFTWARE AND HARDWARE

SOFTWARE:

The software used to run the Python program are:

1. Sublime Text(python)/Windows 8.1 or later
2. MySQL Command Line Client
3. MySQL Workbench 8.0 CE

HARDWARE:

The hardware used to run the project are:

1. 2 GHz Dual Core Processor
2. 4GB Ram
3. 128 GB HDD
4. Monitor with VGA of 1024 x 768 Screen Resolution

PYTHON

Python is a widely-used general-purpose, high-level programming language It was initially designed by Guido Van Rossum in 1991 and developed by the python software foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently. There are two major python versions- python 2 and python 3. Both are quite different.

Reason for increasing popularity:

1.Emphasis on code readability, shorter codes, ease of writing

2.Programmers can express logical concepts in fewer lines of code in comparison to languages such as c++ or java.

3.Python supports multiple programming paradigms, like object-oriented, imperative and functional programming or procedural.

4.There exist inbuilt functions for almost all of the frequently used concepts.

5.Philosophy is "simplicity is the best".

LANGUAGE FEATURES

- Interpreted.
  - There are no separate compilation and execution steps like c and c++.
  - Directly run the program from the source code,
  - Internally, python converts the source code into an intermediate form called bytecodes which is then translated into the native language of the specific computer to run it
  - No need to worry about linking and loading with libraries, etc. Platform independent.
- Platform Independent
  - Python programs can be developed and executed on multiple operatingsystem platforms.
  - Python can be used on Linux, Windows, Macintosh, Solaris, and manymore.

- Free and open source, redistributable.
- High-level language.
  - In python, no need to take care of low-level details such asmanaging the memory used by the program.
- Simple.
  - Closer to the English language; Easy to learn.
  - More emphasis on the solution to the problem rather than the syntax.
- *Embeddable.

    - Python can be used within the c/c++ program to give scriptingcapabilities for the program's users.

- Robust.
    - Exceptional handling features.
    - Memory management techniques inbuilt.
- Rich library support.
    - The python standard library is varied vast.
    - Known as the "batteries included" philosophy of python; It can helpdo various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, email, XML, HTML, wav files, cryptography, GUI, and many more.
    - Besides the standard library, there are various other high-quality Libraries such as the python imaging library which is an amazinglysimple image manipulation library.

DESIGNING AND CODING

LIBRARIES AND MODULES USED:

LIBRARIES USED:

1. tkinter:

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

2. pymysql:

PyMySQL is a driver for connecting to a MySQL database server through the Open Database Connectivity (ODBC) application program interface (API), which is the standard means of connecting to any database.

3. datetime:

Python Datetime module supplies classes to work with date and time. These classes provide several functions to deal with dates, times, and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are manipulating objects and not string or timestamps.

4. **Tkcalendar:**

   tkcalendar is a python module that provides the Calendar and DateEntry widgets for Tkinter. The DateEntry widget is similar to a Combobox, but the drop-down is not a list but a Calendar to select a date.

USER-DEFINED MODULES:

1. **Logo page Module**
2. **Login Module**
3. **Admin Module**
4. **Office Module**
5. **Teacher Module**

FUNCTIONS AND METHODS USED:

i. **connect():**

   To create a connection between the MySQL database and the python application, the connect() method of mysql.connector module is used. Pass the database details like HostName, username, and the database password in the method call. The method returns the connection object.

ii. **cursor():**

   The MySQL Cursor of mysql-connector-python (and similar libraries) is used to execute statements to communicate with the MySQL database. Using the methods of it you can execute SQL statements, fetch data from the result sets, call procedures. You can create a Cursor object using the cursor() method of the Connection object/class.

iii. **execute():**

This method accepts a MySQL query as a parameter and executes the given query.

iv. **fetchall():**

This method retrieves all the rows in the result set of a query and returns them as a list of tuples. (If we execute this after retrieving few rows it returns the remaining ones)

v. **commit():**

The commit() method is used to confirm the changes made by the user to the database. Whenever any change is made to the database using update or any other statements, it is necessary to commit the changes. If we do not use the commit() method after making any changes to the database, the database will not be updated and changes will not be reflected.

vi. **Tk() :**

To create the main window, tkinter offers a method 'Tk()'

vii. **geometry():**

The geometry method is a fundamental one that decides the size, position, and some other attributes of the screen layout we are going to create.

viii. **title():**

The title() function is used to set the title of the widget.

ix. **iconphoto():**

iconphoto sets the icon of the window/frame widget to bitmap. The icon must be an ico type, but not png or jpg type, otherwise, the image will not display as the icon.

x. **resizable():**

the resizable() method is used to allow the Tkinter root window to change its size according to the user's need as well we can prohibit resizing of the Tkinter window. So, if the user wants to create a fixed size window, this method can be used.

xi. **PhotoImage():**

Image can be added with the help of the PhotoImage() method. This is a Tkinter method which means you don't have to import any other module to use it.

xii. **pack():**

The pack() method organizes the widgets in blocks before placing them in the parent widget.

xiii. **askquestion():**

This function is used to ask questions to the user. That has only two options YES or NO.

xiv. **showinfo() :**

This function is used to display some important information.

xv. **exit():**

This function is considered good to be used in production code for the sys module is always available. The optional argument arg can be an integer giving the exit or another type of object. If it is an integer, zero is considered "successful termination".

xvi. **withdraw():**

Tkinter withdraw method hides the window without destroying it internally. It is similar to the iconify method that turns a window into a small icon. Let us suppose we want to reveal the hidden window during the execution of an application then we can use the deiconify() method. It can be invoked with the window or frame of a widget in the application.

xvii. **focus():**

In various applications, tkinter widgets are required to be focused to make them active. Widgets can also grab focus and prevent other events outside the bounds. To manage and give focus to a particular widget, we generally use the focus() method. It focuses on the widget and makes them active until the termination of the program.

xviii. **bind():**

If we bind the press of any key from the keyboard with a function that gets executed. Once the Tkinter GUI window is open, we can press any key on the keyboard and we get a message that the keyboard is pressed.

xix. **Label():**

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

xx. **Button():**

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

xxi. **Entry():**

The Entry widget is used to provide the single-line text-box to the user to accept a value from the user. We can use the Entry widget to accept the text strings from the user. It can only be used for one line of text from the user. For multiple lines of text, we must use the text widget.

xxii. **place():**

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

xxiii. subsample():

Return a new PhotoImage based on the same image as this widget but use only every Xth or Yth pixel.

xxiv. protocol():

Some of the benefits of interfaces and protocols are type hinting during the development process using tools built into IDEs and static type analysis for detection of errors before runtime. This way, a static analysis tool can tell you when you check your code if you're trying to access any members that are not defined on an object, instead of only finding out at runtime.

xxv. mainloop():

mainloop() is an infinite loop used to run the application, wait for an event to occur, and process the event as long as the window is not closed.

xxvi. get():

The get() method returns the value of the item with the specified key.

xxvii. insert():

The Python List insert() method is an inbuilt function in Python that inserts a given element at a given index in a list.

xxviii. destroy():

The destroy() method in Tkinter destroys a widget. It is useful in controlling the behaviour of various widgets which depend on each other. Also, when a process is complete by some user action, we need to destroy the GUI components to free the memory as well as clear the screen. The destroy() method achieves all this.

xxix. **isspace():**

Python String isspace() is a built-in method used for string handling. The isspace() method returns "True" if all characters in the string are whitespace characters, Otherwise, it returns "False". This function is used to check if the argument contains all whitespace characters such as: ' '

xxx. **isalpha():**

Python String isalpha() method is a built-in method used for string handling. The isalpha() methods returns "True" if all characters in the string are alphabetically, Otherwise, it returns "False".

xxxi. **isdigit():**

Python isdigit() function returns a Boolean value TRUE if all the values in the input string are digits; else it returns FALSE.

xxxii. **isalnum():**

isalnum() is a built-in Python function that checks whether all characters in a string are alphanumeric. In other words, isalnum() checks whether a string contains only letters or numbers, or both. If all characters are alphanumeric, isalnum() returns the value True; otherwise, the method returns the value False.

xxxiii. **lower():**

In Python, lower() is a built-in Python method primarily used for string handling. The. The lower() method takes no arguments and returns the lowercase strings from the given string by converting each uppercase character to lowercase. If there are no uppercase characters in the given string, it returns the original string.

xxxiv. **len():**

len() is a built-in function in python. You can use the len() to get the length of the given string, array, list, tuple, dictionary, etc. Value: the given value you want the length of. Return value a return an integer value i.e., the length of the given string, or array, or list, or collections.

xxxv. **int():**

The int() function converts the specified value into an integer number.

xxxvi. **str():**

Python str() function returns the string version of the object.

xxxvii. **tuple():**

The tuple() function is a built-in function in Python that can be used to create a tuple.

xxxviii. **list():**

The list() function is a built-in function in Python that can be used to create a list.

xxxix. **append():**

The append() method in python adds a single item to the existing list.

xl. **Combobox():**

A combobox is a combination of an Entry widget and a Listbox widget. A combobox widget allows you to select one value in a set of values. In addition, it allows you to enter a custom value.

xli. **Treeview():**

A Treeview widget allows you to display data in both tabular and hierarchical structures. To create a Treeview widget, you use the ttk.Treeview class: tree = ttk.Treeview(container, **options) A Treeview widget holds a list of items. Each item has one or more columns.

xlii. **replace():**

The replace() method is a built-in functionality offered in Python programming. It replaces all the occurrences of the old substring with the new substring. Replace() returns a new string in which the old substring is replaced with the new substring.

xliii. **configure():**

configure() is used to configure certain options in widgets. When you create a widget in Tkinter, you have to pass in several values as parameters, used to configure certain features for that Widget.

xliv. **Scrollbar():**

This widget provides a slide controller that is used to implement vertically scrolled widgets, such as Listbox, Text, and Canvas. Note that you can also create horizontal scrollbars on Entry widgets.

xlv. **range():**

The range() function returns a sequence of numbers, starting from 0 by default, increments by 1 (by default), and stops before a specified number.

xlvi. **DateEntry():**

Tk::DateEntry is a drop down widget for selecting dates. It looks like the BrowseEntry widget with an Entry followed by an arrow button, but instead of displaying a Listbox the DateEntry displays a calendar with buttons for each date. The calendar contains buttons for browsing through the months.

xlvii. **Strip():**

Python strip() function is a part of built-in functions available in the Python library. The strip() method removes given characters from start and end of the original string. By default, strip() function removes white spaces from start and end of the string and returns the same string without white spaces.

xlviii. **State():**

This is a python function which helps us to fix the state of the Tkinter window .

xlix. **Style():**

This is a python function used to specify a custom widget style.

l. **Thema_use():**

This helps use to get a current theme, we can use it using the theme_use() method.

li. **Clear():**

The clear() method removes all the elements from a dictionary, list and tuple.

lii. **After():**

After() is a method defined for all tkinter widgets. This method simply calls the function callback after the given delay in ms. If no function is given, it acts similar to time.sleep (but in milliseconds instead of seconds)

liii. **Today():**

Return the current local date.

liv. **Deselect():**

Deselect() is used to remove the check mark from the checkbutton

lv. **Selection():**

Selection in Python Tkinter Treeview returns the tuple of selected items. Selection method returns the row index of selected items in Treeview.

CODE

logo page Module:

```
from tkinter import *


def func():
    logo_win.withdraw()
    import login


logo_win = Tk()
logo_win.geometry('600x350+390+180')
logo_win.overrideredirect(1)
img = PhotoImage(file=r'C:\Users\User\Desktop\New
folder\logo.png')
l = Label(logo_win, image=img).pack()
logo_win.after(3000, func)


logo_win.mainloop()
```

login Module:

```
from tkinter import *

from tkinter import messagebox, ttk

import pymysql as mc

from datetime import datetime


d = e = f = 4

a = b = 0

l1 = []

l2 = []

mycon = mc.connect(host='localhost', user='root', passwd='1822',
database='school')

mycur = mycon.cursor()

mycur.execute('select ut, un, up from login')

data = mycur.fetchall()

fo = open('who.txt','w')

def mainfunc():

    global main, button1, button2, button3, button4

    main = Tk()
    main.geometry('650x400+370+150')
    main.resizable(False, False)
    main.title('Login')
    icon = PhotoImage(file=r'C:\Users\User\Desktop\New folder\office5.png', master=main)
    main.iconphoto(False, icon)
    test = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg1.png', master=main)
    label1 = Label(main, image=test).pack()
    test1 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg2.png', master=main)
    test2 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\admin3.png', master=main)
    test2 = test2.subsample(4, 4)
    test3 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\office4.png', master=main)
    test3 = test3.subsample(4, 4)
    test4 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\teacher2.png', master=main)
    test4 = test4.subsample(4, 4)
    label1 = Label(main, text='Choose the type of login ', font=('Cambria', 15), bg='black', fg='gray').place(x=225, y=1)
    label2 = Label(main, image=test2, height=162, width=162, bg='black').place(x=48, y=80)
    label3 = Label(main, image=test3, height=162, width=162, bg='black').place(x=247, y=80)
    label4 = Label(main, image=test4, height=162, width=162, bg='black').place(x=447, y=80)
    button1 = Button(main, text='Administrator', command=lambda: nad(1), width=150, height=25, font=('Cambria', 12,), bd=3, image=test1, compound=CENTER)
    button1.place(x=50, y=275)
    button1.focus_set()
    button1.bind('<Up>', up_main)
    button1.bind('<Down>', down_main)
    button1.bind('<Return>', nad)
    button2 = Button(main, text='Office', command=lambda: noff(1), width=150, height=25, font=('Cambria', 12), bd=3, image=test1, compound=CENTER)
    button2.place(x=250, y=275)
    button2.bind('<Return>', noff)
    button3 = Button(main, text='Teacher', command=lambda: ntea(1), width=150, height=25, font=('Cambria', 12), bd=3, image=test1, compound=CENTER)
    button3.place(x=450, y=275)
    button3.bind('<Return>', ntea)
    button4 = Button(main, text='Exit', command=lambda: ext(1), width=50, height=20, font=('Cambria', 12), bd=3, image=test1, compound=CENTER)
    button4.place(x=550, y=350)
    button4.bind('<Return>', ext)
    main.bind('<Escape>', ext)
    def close():
        main.withdraw()
        if messagebox.askyesno('','Are you sure to Exit?'):
            messagebox.showinfo('','Thank you have a Great Day!')
        else:
            mainfunc()
    main.protocol('WM_DELETE_WINDOW',close)
    main.mainloop()
def nad(y):
    global flag, login_type
    flag, login_type = 1, 'Administrator login'
    userfunc(login_type)
def noff(y):
    global flag, login_type
    flag, login_type = 2, '        Office login         '
    userfunc(login_type)
def ntea(y):
    global flag, login_type
    flag, login_type = 3, '        Teacher login         '
    userfunc(login_type)
def ext(y):
    main.withdraw()
    messagebox.showinfo('', 'Thank you for using .')
    exit()
def hide():
    global entry2
    if entry2.get() != '':
        pas = entry2.get()
        entry2 = ttk.Entry(win, font=('Cambria', 12, 'bold'), show='●')
        entry2.insert(0, pas)
        entry2.place(x=300, y=254)
        button6 = Button(win, width=25, height=20, borderwidth=0, command=show, image=image1).place(x=500, y=254)
def show():
    global entry2
    if entry2.get() != '':
        pas = entry2.get()
        entry2 = ttk.Entry(win, font=('Cambria', 12, 'bold'))
        entry2.insert(0, pas)
        entry2.place(x=300, y=254)
        button6 = Button(win, width=25, height=20, borderwidth=0,
command=hide, image=image2).place(x=500, y=254)
def up_main(y):
    def up_main1(y):
        def up_main2(y):
            def up_main3(y):
                button1.focus_set()
            button2.focus_set()
            button2.bind('<Up>', up_main3)
        button3.focus_set()
        button3.bind('<Up>', up_main2)
    button4.focus_set()
    button4.bind('<Up>', up_main1)
def down_main(y):
    def down_main1(y):
        def down_main2(y):
            def down_main3(y):
                button1.focus_set()
            button4.focus_set()
            button4.bind('<Down>', down_main3)
        button3.focus_set()
        button3.bind('<Down>', down_main2)
    button2.focus_set()
    button2.bind('<Down>', down_main1)
def down(y):
    def up(y):
        entry1.focus()
    entry2.focus()
    entry2.bind('<Up>', up)
def right(y):
    def left(y):
        button5.focus_set()
    button6.focus_set()
    button6.bind('<Left>', left)
```

```
def enter(y):
    button5.focus_set()
def chance(x):
    if x >= 1:
        main.withdraw()
        if (login_type.replace(' login', '').lower(), entry1.get(), entry2.get()) not in data:
            for i in data:
                l1.append(i[1])
                l2.append(i[2])
            if entry1.get() not in l1:
                label6 = Label(win, text='     incorrect username     ', font=('Cambria', 10, 'bold'), fg='red', bg='black')
                label6.place(x=320, y=228)
                win.after(3000, label6.destroy)
            if entry2.get() not in l2:
                label7 = Label(win, text='     incorrect password     ', font=('Cambria', 10, 'bold'), fg='red', bg='black')
                label7.place(x=320, y=280)
                win.after(3000, label7.destroy)
            label8 = Label(win, text='You have '+str(x)+' chances', font=('Cambria', 10, 'bold'), fg='red', bg='black')
            label8.place(x=335, y=300)
            win.after(3000, label8.destroy)
    else:
        win.withdraw()
        messagebox.showinfo('Info', 'You are an unauthorized user !\nThank you for using .')
        exit()
def sign_in(y):
    global entry1, entry2
    if entry1.get().strip() != '' and entry2.get() != '':
        if (login_type.strip().replace(' login', '').lower(), entry1.get(), entry2.get()) in data:
            win.withdraw()
            messagebox.showinfo('Login info', entry1.get()+' you have logged in successfully .')
            if login_type.strip().replace(' login', '') == 'Administrator':
                import admin
            elif login_type.strip().replace(' login', '') == 'Office':
                import officemain5
            elif login_type.strip().replace(' login', '') == 'Teacher':
                uspa = entry1.get().title()+'\n',entry2.get().title()
                fo.writelines(uspa)
                fo.flush()
                import teacher
        else:
            if flag == 1:
                global d
                d -= 1
                chance(d)
            elif flag == 2:
                global e
                e -= 1
                chance(e)
            elif flag == 3:
                global f
                f -= 1
                chance(f)
    else:
        global a
```

```
        a += 1
        for i in range(1, 5):
            if a == b or (a, b == 1, 2):
                global label5
                label5 = Label(win, text='The required information is empty', font=('Cambria', 10, 'bold'), fg='red', bg='black')
                label5.place(x=215, y=290)
                win.after(2500, label5.destroy)
                break
            else:
                a -= 1
                break
def cancel(y):
    win.withdraw()
    if messagebox.askyesno('', 'Are you sure to Exit ?'):
            mainfunc()
    else:
        userfunc(login_type)
def userfunc(login_type):
    global b, entry1, entry2, win, image1, image2, button5, button6, button7
    b += 1
    main.withdraw()
    win = Tk()
    win.geometry('650x400+370+150')
    win.resizable(False, False)
    win.title(login_type.strip())
    icon1 = PhotoImage(file=r'C:\Users\User\Desktop\New
folder\office5.png', master=win)
    win.iconphoto(False, icon1)
    test = PhotoImage(file=r'C:\Users\User\Desktop\New
folder\bgu1.png', master=win)
    test = test.subsample(1,1)
    label2 = Label(win, image=test).place(x=-2, y=-2)
    test1 = PhotoImage(file=r'C:\Users\User\Desktop\New
folder\bg2.png', master=win)
    label7 = Label(win, text=login_type, font=('Cambria', 15),
bg='black', fg='gray').place(x=250, y=1)
    label3 = Label(win, text='Username :', font=('Cambria', 15, 'bold'),
bg='black', fg='white').place(x=150, y=195)
    entry1 = ttk.Entry(win, font=('Cambria', 12, 'bold'))
    entry1.place(x=300, y=198)
    entry1.focus()
    entry1.bind('<Down>', down)
    entry1.bind('<Return>', down)
    label4 = Label(win, text='Password  :', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=150, y=250)
    entry2 = ttk.Entry(win, show='●', font=('Cambria', 12, 'bold'))
    entry2.place(x=300, y=253)
    entry2.bind('<Return>', enter)
    image1 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\eye.png', master=win)
    image1 = image1.subsample(14, 15)
    image2 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\hide.png', master=win)
    image2 = image2.subsample(14, 15)
    button5 = Button(win, text='Sign in', command=lambda:sign_in(1), width=100, height=25, font=('Cambria', 12), bd=3, image=test1, compound=CENTER)
    button5.place(x=175, y=325)
    button5.bind('<Right>', right)
    button5.bind('<Up>', down)
    button5.bind('<Return>', sign_in)
    button6 = Button(win, text='Exit', command=lambda:cancel(1), width=100, height=25, font=('Cambria', 12), bd=3, image=test1, compound=CENTER)
    button6.place(x=355, y=325)
    button6.bind('<Up>', down)
    button6.bind('<Return>', cancel)
    button7 = Button(win, width=25, height=20, image=image1, borderwidth=0, command=show)
    button7.place(x=500, y=254)
    win.bind('<Escape>', cancel)

    def close():
        win.withdraw()
        mainfunc()
    win.protocol('WM_DELETE_WINDOW',close)
    win.mainloop()
mainfunc()

fo.close()
```

Admin Module:

```
from tkinter import *

from tkinter import messagebox,ttk

import pymysql as mc

from login import *


mycon = mc.connect(host='localhost', user='root', passwd='1822',
database='school')

mycur = mycon.cursor()


def admin():

        global a_win

        a_win = Tk()

        a_win.title('Admin')

        a_win.geometry('650x400+370+150')

        a_win.resizable(False, False)

        i1_a = PhotoImage(file=r'C:\Users\User\Desktop\New
folder\bg22.png', master=a_win)

        l_a = Label(a_win,image=i1_a).pack()
```

```
    i2_a = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg20.png', master=a_win)

    i2_a = i2_a.subsample(1, 1)

    i3_a = PhotoImage(file=r'C:\Users\User\Desktop\New folder\student3.png', master=a_win)

    i3_a = i3_a.subsample(3, 3)

    i4_a = PhotoImage(file=r'C:\Users\User\Desktop\New folder\teacher2.png', master=a_win)

    i4_a = i4_a.subsample(3, 3)

    l1_a = Label(a_win, text='   Admin   ', font=('Cambria', 15), bg='black', fg='gray').place(x=280, y=0)

    l2_a = Label(a_win, image=i3_a, height=200, width=187, bg='black').place(x=50, y=42)

    l3_a = Label(a_win, image=i4_a, height=200, width=187, bg='black').place(x=400, y=42)

    b1_a = Button(a_win, text='Students Details',image=i2_a, compound=CENTER, command=sd, width=182, height=35, font=('Cambria', 12), fg='white', bd=3).place(x=50, y=270)

    b2_a = Button(a_win, text='Teachers Details',image=i2_a, compound=CENTER, command=td, width=182, height=35, font=('Cambria', 12), fg='white', bd=3).place(x=400, y=270)

    def close():

        a_win.withdraw()
```

```
                mainfunc()
        a_win.protocol('WM_DELETE_WINDOW',close)
        a_win.mainloop()
def sd():
    global vsd, tree1
    a_win.withdraw()
    vsd = Tk()
    vsd.title('Students Details')
    vsd.state('zoomed')
    icon01 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\office5.png', master=vsd)
    vsd.iconphoto(False, icon01)
    i1_s = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg27.png', master=vsd)
    l1_s = Label(vsd, image=i1_s).place(x=-3, y=-3)
    frame1 = Frame(vsd, height=300, width=1200, bd=0)
    frame1.pack()
    image6 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg4.png', master=frame1)
    label5 = Label(frame1, image=image6).place(x=-3, y=-3)
    ia2 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\home4.png', master=vsd)

    ia2 = ia2.subsample(3, 3)

    def home_a1():

        vsd.withdraw()

        admin()

    ba1 = Button(vsd, image=ia2, height=30, width=30, bd=0, bg='black', activebackground='black', command=home_a1)

    ba1.place(x=10, y=10)

    l_s = Label(vsd, text='Students Details', font=('Cambria', 15), bg='black', fg='gray').place(x=600,y=3)

    mycur.execute('select * from s_cust1')

    data = mycur.fetchall()

    mycur.execute('select father_name, father_number, mother_name, mother_number, guardian_name, guardian_number, annual_income, 1st_term_fee, 2nd_term_fee from s_cust2')

    data_stu = mycur.fetchall()

    data = list(data)

    for i in range(len(data)):

        for j in range(len(data_stu)):

            data[i] = list(data[i])+list(data_stu[i])
            break
    data2 = []
    s = ttk.Style(vsd)
    s.theme_use('vista')
    s.configure('Treeview', rowheight=31, background='lightblue', foreground='black', font=('Cambria', 10))
    s.configure('Treeview.Heading', font=('Cambria', 10, 'bold'))
    sb2 = ttk.Scrollbar(vsd, orient="vertical")
    sb2.pack(side='right', fill='y')
    col = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17)
    tree1 = ttk.Treeview(vsd, column=col, show='headings', height=12)
    tree1.pack()
    sb1 = ttk.Scrollbar(vsd, orient="horizontal", command=tree1.xview)
    sb1.pack(side='bottom', fill='x')
    tree1.configure(xscrollcommand=sb1.set)
    tree1.configure(yscrollcommand=sb2.set)
    sb2.configure(command=tree1.yview)
    tree1.column(1, anchor=CENTER, stretch=NO, width=100)
    tree1.column(2, anchor=CENTER, stretch=NO, width=120)
```

```
tree1.column(3, anchor=CENTER, stretch=NO, width=150)

tree1.column(4, anchor=CENTER, stretch=NO, width=100)

tree1.column(5, anchor=CENTER, stretch=NO, width=120)

tree1.column(6, anchor=CENTER, stretch=NO, width=120)

tree1.column(7, anchor=CENTER, stretch=NO, width=100)

tree1.column(8, anchor=CENTER, stretch=NO, width=120)

tree1.column(9, anchor=CENTER, stretch=NO, width=150)

tree1.column(10, anchor=CENTER, stretch=NO, width=150)

tree1.column(11, anchor=CENTER, stretch=NO, width=150)

tree1.column(12, anchor=CENTER, stretch=NO, width=150)

tree1.column(13, anchor=CENTER, stretch=NO, width=150)

tree1.column(14, anchor=CENTER, stretch=NO, width=150)

tree1.column(15, anchor=CENTER, stretch=NO, width=120)

tree1.column(16, anchor=CENTER, stretch=NO, width=150)

tree1.column(17, anchor=CENTER, stretch=NO, width=150)

tree1.heading(1, text='Admission.no')

tree1.heading(2, text='Admission date')

tree1.heading(3, text='Name')

tree1.heading(4, text='Class')

tree1.heading(5, text='Date of the Birth')
```

```
tree1.heading(6, text='Gender')
tree1.heading(7, text='Blood group')
tree1.heading(8, text='Mother tongue')
tree1.heading(9, text='Father name')
tree1.heading(10, text='Father phone number')
tree1.heading(11, text='Mother name')
tree1.heading(12, text='Mother phone number')
tree1.heading(13, text='Guardian name')
tree1.heading(14, text='Guardian phone number')
tree1.heading(15, text='Annual income')
tree1.heading(16, text='First Term Fees')
tree1.heading(17, text='Second Term Fees')
for j in range(len(data)):
    for i in data[j]:
        data2.append(i)
    tree1.insert('', END, values=data2)
    data2.clear()
def close():
    vsd.withdraw()
    admin()
    vsd.protocol('WM_DELETE_WINDOW',close)
    search()
    vsd.mainloop()
def td():
    global vtd, tree2
    a_win.withdraw()
    vtd = Tk()
    vtd.title('Teachers Details')
    vtd.state('zoomed')
    icon9 = PhotoImage(file=r'C:\Users\User\Desktop\New
folder\office5.png', master=vtd)
    vtd.iconphoto(False, icon9)
    i1_s = PhotoImage(file=r'C:\Users\User\Desktop\New
folder\bg27.png', master=vtd)
    l1_s = Label(vtd, image=i1_s).place(x=-3, y=-3)
    frame1 = Frame(vtd, height=300, width=1200, bd=0)
    frame1.pack()
    image6 = PhotoImage(file=r'C:\Users\User\Desktop\New
folder\bg4.png', master=frame1)
    ia1 = PhotoImage(file=r'C:\Users\User\Desktop\New
folder\home4.png', master=vtd)
```

```
    ia1 = ia1.subsample(3, 3)

    def home_a():

        vtd.withdraw()

        admin()

    bv1 = Button(vtd, image=ia1, height=30, width=30, bd=0,
bg='black', activebackground='black', command=home_a)

    bv1.place(x=10, y=10)

    label5 = Label(frame1, image=image6).place(x=-3, y=-3)

    l_s = Label(vtd, text='Teachers Details', font=('Cambria', 15),
bg='black', fg='gray').place(x=600,y=3)

    mycur.execute('select * from t_cust')

    data_teach = mycur.fetchall()

    mycur.execute('select qualification, subject, class_teaching1,
class_teaching2, class_teacher, designation, salary_status from
t_cust1')

    data_t = mycur.fetchall()

    data_teach = list(data_teach)

    for i in range(len(data_teach)):

        for j in range(len(data_t)):

            data_teach[i] = list(data_teach[i])+list(data_t[i])

            break
    data_teach2 = []
    style = ttk.Style(vtd)
    style.theme_use('vista')
    style.configure('Treeview', rowheight=31, background='lightblue', foreground='black', font=('Cambria', 10))
    style.configure('Treeview.Heading', font=('Cambria', 10, 'bold'))
    sb2_t = ttk.Scrollbar(vtd, orient="vertical")
    sb2_t.pack(side='right', fill='y')
    col = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)
    tree2 = ttk.Treeview(vtd, column=col, show='headings', height=12)
    tree2.pack()
    sb1_t = ttk.Scrollbar(vtd, orient="horizontal", command=tree2.xview)
    sb1_t.pack(side='bottom', fill='x')
    tree2.configure(xscrollcommand=sb1_t.set)
    tree2.configure(yscrollcommand=sb2_t.set)
    sb2_t.configure(command=tree2.yview)
    tree2.column(1, anchor=CENTER, stretch=NO, width=100)
    tree2.column(2, anchor=CENTER, stretch=NO, width=120)
    tree2.column(3, anchor=CENTER, stretch=NO, width=150)
```

```
tree2.column(4, anchor=CENTER, stretch=NO, width=100)
tree2.column(5, anchor=CENTER, stretch=NO, width=120)
tree2.column(6, anchor=CENTER, stretch=NO, width=120)
tree2.column(7, anchor=CENTER, stretch=NO, width=100)
tree2.column(8, anchor=CENTER, stretch=NO, width=120)
tree2.column(9, anchor=CENTER, stretch=NO, width=150)
tree2.column(10, anchor=CENTER, stretch=NO, width=150)
tree2.column(11, anchor=CENTER, stretch=NO, width=150)
tree2.column(12, anchor=CENTER, stretch=NO, width=150)
tree2.column(13, anchor=CENTER, stretch=NO, width=150)
tree2.column(14, anchor=CENTER, stretch=NO, width=150)
tree2.column(15, anchor=CENTER, stretch=NO, width=150)
tree2.heading(1, text='Teacher Id')
tree2.heading(2, text='Joining date')
tree2.heading(3, text='Name')
tree2.heading(4, text='Date of the Birth')
tree2.heading(5, text='Gender')
tree2.heading(6, text='Blood group')
tree2.heading(7, text='Mother Tongue')
tree2.heading(8, text='Phone.no')
```

```
    tree2.heading(9, text='Qualification')
    tree2.heading(10, text='subject')
    tree2.heading(11, text='Classes Teaching')
    tree2.heading(12, text='Classes Teaching')
    tree2.heading(13, text='Class Teacher Of')
    tree2.heading(14, text='Designation')
    tree2.heading(15, text='Salary Status')
    for j in range(len(data_teach)):
        for i in data_teach[j]:
            data_teach2.append(i)
        tree2.insert('', END, values=data_teach2)
        data_teach2.clear()
    def close():
        vtd.withdraw()
        admin()
    vtd.protocol('WM_DELETE_WINDOW',close)
    search_teach()
    vtd.mainloop()
def search_teach():
    get_l = []
    l=[]

    mycur.execute('select * from t_cust')

    data2 = mycur.fetchall()

    mycur.execute('select qualification, subject, class_teaching1, class_teaching2, class_teacher, designation, salary_status from t_cust1')

    data_t = mycur.fetchall()

    data2 = list(data2)

    for i in range(len(data2)):

        for j in range(len(data_t)):

            data2[i] = list(data2[i])+list(data_t[i])

            break

    def find():

        f3 = 0

        f4 = 0

        append_l = []

        l_dict = {}

        teaching_dict = {}

        for i in tree2.get_children():

            tree2.delete(i)
        get_l = [entry1t.get().strip(), entry2t.get().strip(), combo1t.get().strip(), combo2t.get().strip(), entry3t.get().strip(), entry4t.get().strip(),combo4t.get().strip(), combo5t.get().strip(), combo3t.get().strip()]

        mycur.execute('select teacher_id,name,gender,blood_group,mother_tongue from t_cust')

        data = mycur.fetchall()

        mycur.execute('select subject,class_teaching1,class_teaching2,class_teacher from t_cust1')

        data_t2 = mycur.fetchall()

        data = list(data)

        for i in range(len(data)):

            for j in range(len(data_t2)):

                data[i] = list(data[i])+list(data_t2[i])

                break

        for i in range(len(data)):

            get_l1 = [data[i][6],data[i][7]]

            if combo4t.get().strip() == '' and combo5t.get().strip() == '':

                get_l = [entry1t.get().strip(), entry2t.get().strip(), combo1t.get().strip(), combo2t.get().strip(), entry3t.get().strip(), entry4t.get().strip(),combo4t.get().strip(), combo5t.get().strip(), combo3t.get().strip()]
            elif combo4t.get().strip() == combo5t.get().strip():
                label9t = Label(vtd, text='Enter a valid class.', bg='black', fg='red', font=('Cambria', 10, 'bold'))
                label9t.place(x=1000, y=250)
                vtd.after(1500, label9t.destroy)
            elif combo4t.get().strip() == get_l1[0] or combo5t.get().strip() == get_l1[1]:
                get_l = [entry1t.get().strip(), entry2t.get().strip(), combo1t.get().strip(), combo2t.get().strip(), entry3t.get().strip(), entry4t.get().strip(),combo4t.get().strip(), combo5t.get().strip(), combo3t.get().strip()]
            elif combo4t.get().strip() == get_l1[1] or combo5t.get().strip() == get_l1[0]:
                get_l = [entry1t.get().strip(), entry2t.get().strip(), combo1t.get().strip(), combo2t.get().strip(), entry3t.get().strip(), entry4t.get().strip(),combo5t.get().strip(), combo4t.get().strip(), combo3t.get().strip()]
            for i in get_l:
                if i != '':
                    l_dict[get_l.index(i)] = i
                    index = get_l.index(i)
                    get_l.remove(i)
                    get_l.insert(index, '')
            for i in data:
                for j in l_dict:
                    if i[j] != None and l_dict[j].lower() == i[j].lower():
                        f3 += 1
                    if f3 == len(l_dict):
                        if data2[data.index(i)] not in append_l:
                            append_l.append(data2[data.index(i)])
                        f4 = 1
                    if f3 != len(l_dict) and f4 != 1:
                        for k in tree2.get_children():
                            tree2.delete(k)
                f3 = 0
        for i in append_l:
            tree2.insert('', END, values=i)
    def reset_s():
        entry1t.delete(0, END)
        entry2t.delete(0, END)
        entry3t.delete(0, END)
        entry4t.delete(0, END)
        combo1t.delete(0, END)
```

```
        combo2t.delete(0, END)
        combo3t.delete(0, END)
        combo4t.delete(0, END)
        combo5t.delete(0, END)
        for k in tree2.get_children():
            tree2.delete(k)
        for i in data2:
            tree2.insert('', END, values=i)
    label1t = Label(vtd, text='Teacher id : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=150, y=50)
    entry1t = ttk.Entry(vtd, font=('Cambria', 12, 'bold'))
    entry1t.place(x=330, y=50)
    label2t = Label(vtd, text='Name : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=550, y=50)
    entry2t = ttk.Entry(vtd, font=('Cambria', 12, 'bold'))
    entry2t.place(x=700, y=50)
    label3t = Label(vtd, text='Gender : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=900, y=50)
    list1t = ['Male', 'Female', 'Others']
    combo1t = ttk.Combobox(vtd, value=list1t, font=('Cambria', 12, 'bold'))
    combo1t.place(x=1045, y=50)

    label4t = Label(vtd, text='Blood Gruop : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=150, y=120)

    list2t = ['O+ve', 'O-ve', 'A+ve', 'A-ve', 'B+ve', 'B-ve', 'AB+ve', 'AB-ve', 'A1+ve', 'Hh']

    combo2t = ttk.Combobox(vtd, value=list2t, font=('Cambria', 12, 'bold'))

    combo2t.place(x=330, y=120)

    label5t = Label(vtd, text='Mother Tongue : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=550, y=120)

    entry3t = ttk.Entry(vtd, font=('Cambria', 12, 'bold'))

    entry3t.place(x=700, y=120)

    label6t = Label(vtd, text='Subject : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=900, y=120)

    entry4t = ttk.Entry(vtd, font=('Cambria', 12, 'bold'))

    entry4t.place(x=1045, y=120)

    label7t = Label(vtd, text='Class Teacher Of : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=270, y=200)

    list3t = ['I', 'II', 'III', 'IV', 'V', 'VI', 'VII', 'VIII', 'IX', 'X', 'XI', 'XII']

    combo3t = ttk.Combobox(vtd, value=list3t, font=('Cambria', 12, 'bold'))

    combo3t.place(x=430, y=200)
    label8t = Label(vtd, text='Class Teaching : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=800, y=200)

    list4t = ['I', 'II', 'III', 'IV', 'V', 'VI', 'VII', 'VIII', 'IX', 'X', 'XI', 'XII']

    combo4t = ttk.Combobox(vtd, value=list4t, font=('Cambria', 12, 'bold'), width=5, height=8)

    combo4t.place(x=950, y=200)

    combo5t = ttk.Combobox(vtd, value=list4t, font=('Cambria', 12, 'bold'), width=5, height=8)

    combo5t.place(x=1100, y=200)

    image9 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg20.png', master=vtd)

    image9 = image9.subsample(2, 2)

    button1t = Button(vtd, text='Search', font=('Cambria', 12, 'bold'), image=image9, height=25, width=100, bg='black', fg='white', compound=CENTER, command=find)

    button1t.place(x=500, y=250)

    button1t.image = image9

    button2t = Button(vtd, text='Reset', font=('Cambria', 12, 'bold'), image=image9, height=25, width=100, bg='black', fg='white', compound=CENTER, command=reset_s)

    button2t.place(x=750, y=250)

    button2t.image = image9
```

```
def search():
    get_l = []
    l=[]
    mycur.execute('select * from s_cust1')
    data2 = mycur.fetchall()
    mycur.execute('select father_name, father_number, mother_name,
mother_number, guardian_name, guardian_number, annual_income,
1st_term_fee, 2nd_term_fee from s_cust2')
    data_stu = mycur.fetchall()
    data2 = list(data2)
    for i in range(len(data2)):
        for j in range(len(data_stu)):
            data2[i] = list(data2[i])+list(data_stu[i])
            break
    def find():
        f = 0
        f2 = 0
        l_dict = {}
        for i in tree1.get_children():
            tree1.delete(i)
```

```
get_l = [entry1s.get().strip(), entry2s.get().strip(), combo1s.get().strip(), combo2s.get().strip(), combo3s.get().strip(), entry3s.get().strip(), combo4s.get().strip(), combo5s.get().strip()]

mycur.execute('select admission_no,name,class,gender,blood_group,mother_tongue from s_cust1')

data = mycur.fetchall()

mycur.execute('select 1st_term_fee, 2nd_term_fee from s_cust2')

data_stu = mycur.fetchall()

data = list(data)

for i in range(len(data)):

    for j in range(len(data_stu)):

        data[i] = list(data[i])+list(data_stu[i])

        break

for i in get_l:

    if i != '':

        l_dict[get_l.index(i)] = i

        index = get_l.index(i)

        get_l.remove(i)

        get_l.insert(index, '')

for i in data:
        for j in l_dict:
            if str(l_dict[j]).lower() == str(i[j]).lower():
                f += 1
            if f == len(l_dict):
                tree1.insert('', END, values=data2[data.index(i)])
                f2 = 1
            if f != len(l_dict) and f2 != 1:
                for k in tree1.get_children():
                    tree1.delete(k)
        f = 0


def reset_s():
    entry1s.delete(0, END)
    entry2s.delete(0, END)
    combo1s.delete(0, END)
    combo2s.delete(0, END)
    combo3s.delete(0, END)
    entry3s.delete(0, END)
    combo4s.delete(0, END)
    combo5s.delete(0, END)
```

```
        for k in tree1.get_children():
            tree1.delete(k)
        for i in data2:
            tree1.insert('', END, values=i)
    label1s = Label(vsd, text='Admission Number : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=150, y=50)
    entry1s = ttk.Entry(vsd, font=('Cambria', 12, 'bold'))
    entry1s.place(x=330, y=50)
    label2s = Label(vsd, text='Name : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=550, y=50)
    entry2s = ttk.Entry(vsd, font=('Cambria', 12, 'bold'))
    entry2s.place(x=680, y=50)
    label3s = Label(vsd, text='Class : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=895, y=50)
    list1s = ['I', 'II', 'III', 'IV', 'V', 'VI', 'VII', 'VIII', 'IX', 'X', 'XI', 'XII']
    combo1s = ttk.Combobox(vsd, value=list1s, font=('Cambria', 12, 'bold'))
    combo1s.place(x=1045, y=50)
    label4s=Label(vsd, text='Gender : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=150, y=120)
    list2s = ['Male', 'Female', 'Others']
    combo2s = ttk.Combobox(vsd, value=list2s, font=('Cambria', 12, 'bold'))
    combo2s.place(x=330, y=120)
    label5s = Label(vsd, text='Blood Group : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=550, y=120)
    list3s = ['O+ve', 'O-ve', 'A+ve', 'A-ve', 'B+ve', 'B-ve', 'AB+ve', 'AB-ve', 'A1+ve', 'Hh']
    combo3s = ttk.Combobox(vsd, value=list3s, font=('Cambria', 12, 'bold'))
    combo3s.place(x=680, y=120)
    label6s = Label(vsd, text='Mother Tongue : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=895, y=120)
    entry3s = ttk.Entry(vsd, font=('Cambria', 12, 'bold'))
    entry3s.place(x=1045, y=120)
    label7s = Label(vsd, text='Term I : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=270, y=200)
    list4s = ['Paid', 'Partially Paid', 'Not Paid']
    combo4s = ttk.Combobox(vsd, value=list4s, font=('Cambria', 12, 'bold'))
    combo4s.place(x=350, y=200)
    label8s = Label(vsd, text='Term II : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=800, y=200)
    combo5s = ttk.Combobox(vsd, value=list4s, font=('Cambria', 12, 'bold'))

    combo5s.place(x=893, y=200)

    image9 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg20.png', master=vsd)

    image9 = image9.subsample(2, 2)

    button1s = Button(vsd, text='Search', font=('Cambria', 12, 'bold'), image=image9, height=25, width=100, bg='black', fg='white', compound=CENTER, command=find)

    button1s.place(x=500, y=250)

    button1s.image = image9

    button2s = Button(vsd, text='Reset', font=('Cambria', 12, 'bold'), image=image9, height=25, width=100, bg='black', fg='white', compound=CENTER, command=reset_s)

    button2s.place(x=750, y=250)

    button2s.image = image9


admin()
```

Teacher Module:

```
from tkinter import *

from tkinter import messagebox, ttk

import pymysql as mc

from login import *


mycon = mc.connect(host='localhost', user='root', passwd='1822',
database='school')

mycur = mycon.cursor()

fo = open('who.txt','r+')

unpa = fo.readlines()

if len(unpa) != 0:

        un = unpa[0].replace('\n','')

        pa = unpa[1]

        mycur.execute('select Class from login where up = "'+pa+'"')

        tdata1 = mycur.fetchall()

        Class = tdata1[0][0]

        def teacher():

                global t_win
                t_win = Tk()
                t_win.title('Teacher')
                t_win.geometry('650x400+370+150')
                t_win.resizable(False, False)
                icon3 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\office5.png', master=t_win)
        t_win.iconphoto(False, icon3)
                i1 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg22.png', master=t_win)
                i2 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg4.png', master=t_win)
                i2 = i2.subsample(2, 2)
                l1 = Label(t_win, image=i1).place(x=-2,y=-1)
                l2 = Label(t_win, text='Teacher', font=('Cambria', 15), bg='black', fg='gray').place(x=300, y=1)
                l3 = Label(t_win, text='Teacher Name : ', font=('Cambria', 13), bg='black', fg='deep sky blue').place(x=25, y=40)
                l4 = Label(t_win, text=un, font=('Cambria', 13), bg='black', fg='violet red').place(x=145, y=40)
                l5 = Label(t_win, text='Class : ', font=('Cambria', 13), bg='black', fg='violet red').place(x=540, y=40)
                l6 = Label(t_win, text=Class, font=('Cambria', 13), bg='black', fg='deep sky blue').place(x=595, y=40)

                b1 = Button(t_win, image=i2, text='Add students mark', compound=CENTER, width=182, height=35, command=add, font=('Cambria', 12), fg='white', bg='black', activebackground='black', activeforeground='white', bd=3)

                b1.place(x=50, y=285)

                b2 = Button(t_win, image=i2, text='Update students mark', compound=CENTER, width=182, height=35, command=update, font=('Cambria', 12), fg='white', bg='black', activebackground='black', activeforeground='white', bd=3)

                b2.place(x=230, y=150)

                b3 = Button(t_win, image=i2, text='View students mark', compound=CENTER, width=182, height=35, command=view, font=('Cambria', 12), fg='white', bg='black', activebackground='black', activeforeground='white', bd=3)

                b3.place(x=410, y=285)

                def close1():

                    t_win.withdraw()

                    mainfunc()

                t_win.protocol('WM_DELETE_WINDOW',close1)

                t_win.mainloop()
```

```
    def add():
        mycur.execute('select admission_no, name, class, pa1,
pa2, pa3, pa4 from class_det where class = "'+Class+'"')
        data = mycur.fetchall()
        data2 =[]
        t_win.withdraw()
        add_m = Tk()
        add_m.title('Add students Mark')
        add_m.state('zoomed')
        icon4 = PhotoImage(file=r'C:\Users\User\Desktop\New
folder\office5.png', master=add_m)
    add_m.iconphoto(False, icon4)
        f = Frame(add_m, height=326, width=1200).pack()
        i3 = PhotoImage(file=r'C:\Users\User\Desktop\New
folder\bg20.png', master=add_m)
        ia1 = PhotoImage(file=r'C:\Users\User\Desktop\New
folder\bg24.png', master=add_m)
        ia2 = PhotoImage(file=r'C:\Users\User\Desktop\New
folder\bg4.png', master=add_m)
        ia3 = PhotoImage(file=r'C:\Users\User\Desktop\New
folder\home4.png', master=add_m)
        ia3 = ia3.subsample(3, 3)
        la1 = Label(add_m, image=ia1).place(x=-3, y=-3)
        la2 = Label(add_m, image=ia2, width=1150, height=350).place(x=100,y=45)
        la3 = Label(add_m, text=' Add Students Mark ', font=('Cambria', 15), bg='black', fg='gray').place(x=612,y=0)
        def home():
            add_m.withdraw()
            teacher()
        ba2 = Button(add_m, image=ia3, height=30, width=30, bd=0, bg='black', activebackground='black', command=home)
        ba2.place(x=10, y=10)
        s = ttk.Style(add_m)
        s.theme_use('vista')
        s.configure('Treeview', rowheight=29, background='lightblue', foreground='black', font=('Cambria', 10))
        s.configure('Treeview.Heading', font=('Cambria', 10, 'bold'))
        sb1 = ttk.Scrollbar(add_m, orient="vertical")
        sb1.pack(side='right', fill='y')
        col = (1, 2, 3, 4, 5, 6, 7)
        tree1 = ttk.Treeview(add_m, column=col, show='headings', height=12)
```

```
        tree1.pack()
        tree1.configure(yscrollcommand=sb1.set)
        sb1.configure(command=tree1.yview)
        sb2 = ttk.Scrollbar(add_m, orient="horizontal",
command=tree1.xview)
        sb2.pack(side='bottom', fill='x')
        tree1.configure(xscrollcommand=sb2.set)
        tree1.column(1, anchor=CENTER, stretch=NO,
width=100)
        tree1.column(2, anchor=CENTER, stretch=NO,
width=150)
        tree1.column(3, anchor=CENTER, stretch=NO, width=80)
        tree1.column(4, anchor=CENTER, stretch=NO,
width=310)
        tree1.column(5, anchor=CENTER, stretch=NO,
width=310)
        tree1.column(6, anchor=CENTER, stretch=NO,
width=310)
        tree1.column(7, anchor=CENTER, stretch=NO,
width=310)
        tree1.heading(1, text='Admission.no')
        tree1.heading(2, text='Name')
        tree1.heading(3, text='Class')
        tree1.heading(4, text='PA1 Mark')
        tree1.heading(5, text='PA2 Mark')
        tree1.heading(6, text='PA3 Mark')
        tree1.heading(7, text='PA4 Mark')
        for j in range(len(data)):
            for i in data[j]:
                data2.append(i)
            tree1.insert('', END, values=data2)
            data2.clear()
        mycur.execute('select admission_no,pa1,pa2,pa3,pa4,class from class_det')
        tdata2 = mycur.fetchall()
        list_1 = []
        list_2 = ['PA1', 'PA2', 'PA3', 'PA4']
        list_3 = []
        a = 0
        for i in tdata2:
            if (i[1] == None or i[2] == None or i[3] == None  or i[4] == None) and i[5] == Class:
                a = 1
                list_1.append(i[0])
        if a == 0:
            la6 = Label(add_m, text=' All marks are added to the
students ', font=('Cambria', 12, 'bold'), bg='black', fg='red')
            la6.place(x=550, y=200)
        if a == 1:
            def add_mark():
                global ex, ad
                ad = ca1.get().strip()
                ex = ca2.get().strip()
                if ca1.get().strip() != '' or ca1.get().strip() != '' :
                    if ca1.get().strip().isdigit() :
                        if int(ca1.get().strip()) in list_1 and
ca2.get().strip() in list_2:
                            for i in tdata2:
                                if int(ca1.get().strip()) in
i:
                                    ind = tdata2.index(i)
        if
tdata2[ind][tdata2[ind].index(tdata2[ind][(int(ca2.get().strip()[-
1:]))])]] == None:
    la4.destroy()
    la5.destroy()
    ca1.destroy()
    ca2.destroy()
    ba1.destroy()
    if Class in ['XI','XII']:
        subject = (' English : ', ' Maths : ', ' Chemistry : ', ' Physics : ', ' Computer Science : ')
        sub = (' Eng : ', ' Mat : ', ' Chem : ', ' Phy : ', ' C.S : ')
    else:
        subject = (' English : ', ' Maths : ', ' Science : ', ' Social Science : ', ' 2nd Language : ')
        sub = (' Eng : ', ' Mat : ', ' Sci : ', ' Sco : ', ' IILang : ')
        def save():
            if en1.get().strip().isdigit() and en2.get().strip().isdigit() and en3.get().strip().isdigit() and en4.get().strip().isdigit() and en5.get().strip().isdigit():
            if 0 <= int(en1.get().strip()) <= 100 and 0 <= int(en2.get().strip()) <= 100 and 0 <= int(en3.get().strip()) <= 100 and 0 <= int(en4.get().strip()) <= 100 and 0 <= int(en5.get().strip()) <= 100 :
```

```
        code_a1 = 'update class_det set '+ex+' =
"'+sub[0]+en1.get().strip()+'  '+sub[1]+en2.get().strip()+'
'+sub[2]+en3.get().strip()+'  '+sub[3]+en4.get().strip()+'
'+sub[4]+en5.get().strip()+'" where admission_no = '+ad
mycur.execute(code_a1)
mycon.commit()
add_m.withdraw()
    if messagebox.showinfo('Info','student marks successfully
saved .'):
    add()
    else:
        la11 = Label(add_m, text='Marks should be between 0 to
100 !', font=('Cambria', 10, 'bold'), bg='black', fg='red')

    la11.place(x=590, y=240)
    if int(en1.get().strip()) < 0 or int(en1.get().strip()) > 100 :

    en1.delete(0, END)
                                                            if
int(en2.get().strip()) < 0 or int(en2.get().strip()) > 100 :

    en2.delete(0, END)
    if  int(en3.get().strip()) < 0 or int(en3.get().strip()) > 100 :
```

```
    en3.delete(0, END)

    if  int(en4.get().strip()) < 0 or int(en4.get().strip()) > 100 :

        en4.delete(0, END)

    if  int(en5.get().strip()) < 0 or int(en5.get().strip()) > 100:
        en5.delete(0, END)

        add_m.after(2500, la11.destroy)

    else:

        la12 = Label(add_m, text=' Marks should be integers !',
font=('Cambria', 10, 'bold'), bg='black', fg='red')

    la12.place(x=615, y=240)

    if  en1.get().strip().isalpha :

        en1.delete(0, END)

    if  en2.get().strip().isalpha :

        en2.delete(0, END)

    if  en3.get().strip().isalpha :

        en3.delete(0, END)

    if  en4.get().strip().isalpha :

        en4.delete(0, END)

    if en5.get().strip().isalpha :
```

```
    en5.delete(0, END)
    add_m.after(2500, la12.destroy)

def back():

    add_m.withdraw()

    add()

    la6 = Label(add_m, text=subject[0], font=('Cambria', 15, 'bold'),
bg='black', fg='white')

    la6.place(x=150, y=65)

    en1 = ttk.Entry(add_m, font=('Cambria', 15, 'bold'))

    en1.place(x=300, y=65)

    la7 = Label(add_m, text=subject[1], font=('Cambria', 15, 'bold'),
bg='black', fg='white')

    la7.place(x=850, y=65)

    en2 = ttk.Entry(add_m, font=('Cambria', 15, 'bold'))

    en2.place(x=950, y=65)

    la8 = Label(add_m, text=subject[2], font=('Cambria', 15, 'bold'),
bg='black', fg='white')

    la8.place(x=150, y=250)

    en3 = ttk.Entry(add_m, font=('Cambria', 15, 'bold'))

    en3.place(x=300, y=250)
```

```
la9 = Label(add_m, text=subject[3], font=('Cambria', 15, 'bold'), bg='black', fg='white')

la9.place(x=850, y=250)

en4 = ttk.Entry(add_m, font=('Cambria', 15, 'bold'))

en4.place(x=950, y=250)

la10 = Label(add_m, text=subject[4], font=('Cambria', 15, 'bold'), bg='black', fg='white')

la10.place(x=490, y=160)

en5 = ttk.Entry(add_m, font=('Cambria', 15, 'bold'))

en5.place(x=690, y=160)

ba3 = Button(add_m, image=i3, text='Save', compound=CENTER, command=save, width=100, height=25, font=('Cambria', 12), fg='white', bg='black', activebackground='black', activeforeground='white', bd=3)

ba3.place(x=580,y=280)

ba4 = Button(add_m, image=i3, text='Back', compound=CENTER, command=back, width=100, height=25, font=('Cambria', 12), fg='white', bg='black', activebackground='black', activeforeground='white', bd=3)

ba4.place(x=710,y=280)

else:
```

```
                                la13 = Label(add_m, text=' Marks are added for this Exam !', font=('Cambria', 10, 'bold'), bg='black', fg='red')
                                la13.place(x=600, y=220)
                                add_m.after(2500, la13.destroy)
                    else:
                        la14 = Label(add_m, text=' Invalid Admission.no !', font=('Cambria', 10, 'bold'), bg='black', fg='red')
                        la14.place(x=618, y=220)
                        add_m.after(2500, la14.destroy)
                        ca2.delete(0, END)
                else:
                    la15 = Label(add_m, text=' Required information is Empty !', font=('Cambria', 10, 'bold'), bg='black', fg='red')
                    la15.place(x=600, y=230)
                    add_m.after(2500, la15.destroy)
                    ca2.delete(0, END)
            la4 = Label(add_m, text=' Admission_no : ', font=('Cambria', 15, 'bold'), bg='black', fg='white')
            la4.place(x=230, y=180)
            ca1 = ttk.Combobox(add_m, value=list_1, font=('Cambria', 15, 'bold'), width=5, height=10)

            ca1.place(x=400, y=180)

            la5 = Label(add_m, text=' Exam : ', font=('Cambria', 15, 'bold'), bg='black', fg='white')

            la5.place(x=900, y=180)

            ca2 = ttk.Combobox(add_m, value=list_2, font=('Cambria', 15, 'bold'), width=5, height=10)

            ca2.place(x=1000, y=180)

            ba1 = Button(add_m, image=i3, text='Add Marks', compound=CENTER, command=add_mark, width=100, height=30, font=('Cambria', 12), fg='white', bg='black', activebackground='black', activeforeground='white', bd=3)

            ba1.place(x=640,y=270)

        def close2():

            add_m.withdraw()

            teacher()

        add_m.protocol('WM_DELETE_WINDOW',close2)

        add_m.mainloop()

    def view():

        mycur.execute('select admission_no, name, class, pa1, pa2, pa3, pa4 from class_det where class = "'+Class+'"')
        data = mycur.fetchall()
        data2 =[]
        t_win.withdraw()
        view_m = Tk()
        view_m.title('View Students Marks')
        view_m.state('zoomed')
        icon5 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\office5.png', master=view_m)
    view_m.iconphoto(False, icon5)
        iv1 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\home4.png', master=view_m)
        iv1 = iv1.subsample(3, 3)
        iv2 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg26.png', master=view_m)
        l = Label(view_m, image=iv2)
        l.place(x=-3, y=-3)
        l.image = iv2
        lv1 = Label(view_m, text='View Students Mark', font=('Cambria', 15), bg='black', fg='gray').pack()
        f2 = Frame(view_m,height=20, bg='Black').pack()
        def home_v():
            view_m.withdraw()
            teacher()
        bv1 = Button(view_m, image=iv1, height=30, width=30,
bd=0, bg='black', activebackground='black', command=home_v)
        bv1.place(x=10, y=10)
        bv1.image = iv1
        s = ttk.Style(view_m)
        s.theme_use('vista')
        s.configure('Treeview', rowheight=24,
background='lightblue', foreground='black', font=('Cambria', 10))
        s.configure('Treeview.Heading', font=('Cambria', 10,
'bold'))
        col = (1, 2, 3, 4, 5, 6, 7)
        tree4 = ttk.Treeview(view_m, column=col,
show='headings', height=26)
        tree4.pack()
        sb5 = ttk.Scrollbar(view_m, orient="horizontal",
command=tree4.xview)
        sb5.pack(side='bottom', fill='x')
        tree4.configure(xscrollcommand=sb5.set)
        tree4.column(1, anchor=CENTER, stretch=NO,
width=100)
```

```
        tree4.column(2, anchor=CENTER, stretch=NO,
width=150)
        tree4.column(3, anchor=CENTER, stretch=NO, width=80)
        tree4.column(4, anchor=CENTER, stretch=NO,
width=310)
        tree4.column(5, anchor=CENTER, stretch=NO,
width=310)
        tree4.column(6, anchor=CENTER, stretch=NO,
width=310)
        tree4.column(7, anchor=CENTER, stretch=NO,
width=310)
        tree4.heading(1, text='Admission.no')
        tree4.heading(2, text='Name')
        tree4.heading(3, text='Class')
        tree4.heading(4, text='PA1 Mark')
        tree4.heading(5, text='PA2 Mark')
        tree4.heading(6, text='PA3 Mark')
        tree4.heading(7, text='PA4 Mark')
        for j in range(len(data)):
            for i in data[j]:
                data2.append(i)
            tree4.insert('', END, values=data2)
            data2.clear()
        def close3():
            view_m.withdraw()
            teacher()
        view_m.protocol('WM_DELETE_WINDOW',close3)
    def update():
        global up_m
        mycur.execute('select admission_no, name, class, pa1,
pa2, pa3, pa4 from class_det where class = "'+Class+'"')
        data1 = mycur.fetchall()
        data3 =[]
        t_win.withdraw()
        up_m = Tk()
        up_m.title('Update students Mark')
        up_m.state('zoomed')
        icon6 = PhotoImage(file=r'C:\Users\User\Desktop\New
folder\office5.png', master=up_m)
    up_m.iconphoto(False, icon6)
        f1 = Frame(up_m, height=326, width=1200).pack()
        iu1 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg24.png', master=up_m)
        iu2 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg4.png', master=up_m)
        iu3 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\home4.png', master=up_m)
        iu3 = iu3.subsample(3, 3)
        lu1 = Label(up_m, image=iu1)
        lu1.place(x=-3, y=-3)
        lu1.image = iu1
        lu2 = Label(up_m, image=iu2, width=1150, height=350)
        lu2.place(x=100,y=45)
        lu2.image = iu2
        lu3 = Label(up_m, text=' Update Students Mark ', font=('Cambria', 15), bg='black', fg='gray').place(x=580,y=0)
        s1 = ttk.Style(up_m)
        s1.theme_use('vista')
        s1.configure('Treeview', rowheight=29, background='lightblue', foreground='black', font=('Cambria', 10))
        s1.configure('Treeview.Heading', font=('Cambria', 10, 'bold'))
        sb3 = ttk.Scrollbar(up_m, orient="vertical")
        sb3.pack(side='right', fill='y')
        col1 = (1, 2, 3, 4, 5, 6, 7)
        tree3 = ttk.Treeview(up_m, column=col1, show='headings', height=12)
        tree3.pack()
        tree3.configure(yscrollcommand=sb3.set)
        sb3.configure(command=tree3.yview)
        sb4 = ttk.Scrollbar(up_m, orient="horizontal", command=tree3.xview)
        sb4.pack(side='bottom', fill='x')
        tree3.configure(xscrollcommand=sb4.set)
        tree3.column(1, anchor=CENTER, stretch=NO, width=100)
        tree3.column(2, anchor=CENTER, stretch=NO, width=150)
        tree3.column(3, anchor=CENTER, stretch=NO, width=80)
        tree3.column(4, anchor=CENTER, stretch=NO, width=310)
        tree3.column(5, anchor=CENTER, stretch=NO, width=310)
        tree3.column(6, anchor=CENTER, stretch=NO, width=310)
        tree3.column(7, anchor=CENTER, stretch=NO,
width=310)
        tree3.heading(1, text='Admission.no')
        tree3.heading(2, text='Name')
        tree3.heading(3, text='Class')
        tree3.heading(4, text='PA1 Mark')
        tree3.heading(5, text='PA2 Mark')
        tree3.heading(6, text='PA3 Mark')
        tree3.heading(7, text='PA4 Mark')
        for j in range(len(data1)):
            for i in data1[j]:
                data3.append(i)
            tree3.insert('', END, values=data3)
            data3.clear()
        def home1():
            up_m.withdraw()
            teacher()
        mycur.execute('select admission_no,pa1,pa2,pa3,pa4
from class_det where class = "'+Class+'"')
        tdata3 = mycur.fetchall()
        list_4 = list()
        for i in tdata3:
            list_4.append(i[0])
        list_5 = ['PA1', 'PA2', 'PA3', 'PA4']
        def up_mark():
            global ex, ad
            ad = cu1.get().strip()
            ex = cu2.get().strip()
            if cu1.get().strip() != '' or cu2.get().strip() != '' :
                if cu1.get().strip().isdigit() :
                    if int(cu1.get().strip()) in list_4 and
cu2.get().strip() in list_5:
                        for i in tdata3:
                            if int(cu1.get().strip()) in i:
                                ind1 = tdata3.index(i)
                        lu4.destroy()
                        lu5.destroy()
                        cu1.destroy()
                        cu2.destroy()
                        bu1.destroy()
                        if Class in ['XI','XII']:
                            subject1 = (' English : ', ' Maths
: ', ' Chemistry : ', ' Physics : ', ' Computer Science : ')

                            sub1 = (' Eng : ', ' Mat : ', ' Chem
: ', ' Phy : ', ' C.S : ')

                        else:

                            subject1 = (' English : ', ' Maths
: ', ' Science : ', ' Social Science : ', ' 2nd Language : ')

                            sub1 = (' Eng : ', ' Mat : ', ' Sci : ',
' Sco : ', ' IILang : ')

                        def save1():

                            if enu1.get().strip().isdigit()
and enu2.get().strip().isdigit() and enu3.get().strip().isdigit() and
enu4.get().strip().isdigit() and enu5.get().strip().isdigit():

                                if 0 <=
int(enu1.get().strip()) <= 100 and 0 <= int(enu2.get().strip()) <=
100 and 0 <= int(enu3.get().strip()) <= 100 and 0 <=
int(enu4.get().strip()) <= 100 and 0 <= int(enu5.get().strip()) <=
100 :

                                    code_u1 = 'update
class_det set '+ex+' = "'+sub1[0]+enu1.get().strip()+'
'+sub1[1]+enu2.get().strip()+'  '+sub1[2]+enu3.get().strip()+'
'+sub1[3]+enu4.get().strip()+'  '+sub1[4]+enu5.get().strip()+'"
where admission_no = '+ad
    mycur.execute(code_u1)

                                    mycon.commit()
                                    up_m.withdraw()
                                    if
messagebox.showinfo('Info','student marks successfully saved .'):
                                        update()
                                else:
                                    lu11 = Label(up_m,
text='Marks should be between 0 to 100 !', font=('Cambria', 10,
'bold'), bg='black', fg='red')
                                    lu11.place(x=590,
y=240)
                                    if
int(enu1.get().strip()) < 0 or int(enu1.get().strip()) > 100 :
                                        enu1.delete(0,
END)
                                    if
int(enu2.get().strip()) < 0 or int(enu2.get().strip()) > 100 :
                                        enu2.delete(0,
END)
                                    if
int(enu3.get().strip()) < 0 or int(enu3.get().strip()) > 100 :
                                        enu3.delete(0,
END)
```

```
                                    if
int(enu4.get().strip()) < 0 or int(enu4.get().strip()) > 100 :
                                        enu4.delete(0,
END)
                                    if
int(enu5.get().strip()) < 0 or int(enu5.get().strip()) > 100 :
                                        enu5.delete(0,
END)
                                    up_m.after(2500,
lu11.destroy)
                            else:
                                lu12 = Label(up_m, text='
Marks should be integers !', font=('Cambria', 10, 'bold'), bg='black',
fg='red')
                                lu12.place(x=615, y=240)
                                if
enu1.get().strip().isalpha :
                                    enu1.delete(0, END)
                                if
enu2.get().strip().isalpha :
                                    enu2.delete(0, END)
                                if
enu3.get().strip().isalpha :
```

```
                                    enu3.delete(0, END)
                                if
enu4.get().strip().isalpha :
                                    enu4.delete(0, END)
                                if
enu5.get().strip().isalpha :
                                    enu5.delete(0, END)
                                up_m.after(2500,
lu12.destroy)
                        def back1():
                            up_m.withdraw()
                            update()
                        lu6 = Label(up_m, text=subject1[0],
font=('Cambria', 15, 'bold'), bg='black', fg='white')
                        lu6.place(x=150, y=65)
                        enu1 = ttk.Entry(up_m,
font=('Cambria', 15, 'bold'))
                        enu1.place(x=300, y=65)
                        lu7 = Label(up_m, text=subject1[1],
font=('Cambria', 15, 'bold'), bg='black', fg='white')
                        lu7.place(x=850, y=65)
                        enu2 = ttk.Entry(up_m,
font=('Cambria', 15, 'bold'))
                        enu2.place(x=950, y=65)
                        lu8 = Label(up_m, text=subject1[2],
font=('Cambria', 15, 'bold'), bg='black', fg='white')
                        lu8.place(x=150, y=250)
                        enu3 = ttk.Entry(up_m,
font=('Cambria', 15, 'bold'))
                        enu3.place(x=300, y=250)
                        lu9 = Label(up_m, text=subject1[3],
font=('Cambria', 15, 'bold'), bg='black', fg='white')
                        lu9.place(x=850, y=250)
                        enu4 = ttk.Entry(up_m,
font=('Cambria', 15, 'bold'))
                        enu4.place(x=950, y=250)
                        lu10 = Label(up_m, text=subject1[4],
font=('Cambria', 15, 'bold'), bg='black', fg='white')
                        lu10.place(x=490, y=160)
                        enu5 = ttk.Entry(up_m,
font=('Cambria', 15, 'bold'))
                        enu5.place(x=690, y=160)
```

```
                    bu3 = Button(up_m, image=i3, text='Save', compound=CENTER, command=save1, width=100, height=25, font=('Cambria', 12), fg='white', bg='black', activebackground='black', activeforeground='white', bd=3)

                    bu3.place(x=580,y=280)

                    bu3.image = i3

                    bu4 = Button(up_m, image=i3, text='Back', compound=CENTER, command=back1, width=100, height=25, font=('Cambria', 12), fg='white', bg='black', activebackground='black', activeforeground='white', bd=3)

                    bu4.place(x=710,y=280)

                    bu4.image = i3

            else:

                lu14 = Label(up_m, text=' Invalid Admission.no !', font=('Cambria', 10, 'bold'), bg='black', fg='red')

                lu14.place(x=618, y=220)

                up_m.after(2500, lu14.destroy)

                cu2.delete(0, END)

        else:

            lu15 = Label(up_m, text=' Required information is Empty !', font=('Cambria', 10, 'bold'), bg='black', fg='red')

            lu15.place(x=600, y=230)
            up_m.after(2500, lu15.destroy)
        i3 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg20.png', master=up_m)
        lu4 = Label(up_m, text=' Admission_no : ', font=('Cambria', 15, 'bold'), bg='black', fg='white')
        lu4.place(x=230, y=180)
        cu1 = ttk.Combobox(up_m, value=list_4, font=('Cambria', 15, 'bold'), width=5, height=10)
        cu1.place(x=400, y=180)
        lu5 = Label(up_m, text=' Exam : ', font=('Cambria', 15, 'bold'), bg='black', fg='white')
        lu5.place(x=900, y=180)
        list_5 = ['PA1', 'PA2', 'PA3', 'PA4']
        cu2 = ttk.Combobox(up_m, value=list_5, font=('Cambria', 15, 'bold'), width=5, height=10)
        cu2.place(x=1000, y=180)
        bu1 = Button(up_m, image=i3, text='Update Marks', compound=CENTER, command=up_mark, width=100, height=30, font=('Cambria', 12), fg='white', bg='black', activebackground='black', activeforeground='white', bd=3)
        bu1.place(x=640,y=270)
        bu2 = Button(up_m, image=iu3, height=30, width=30, bd=0, bg='black', activebackground='black', command=home1)
        bu2.place(x=10, y=10)
        bu2.image = iu3
        def close3():
            up_m.withdraw()
            teacher()
        up_m.protocol('WM_DELETE_WINDOW',close3)
        up_m.mainloop()
    def non_classteacher():
        global t1_win
        t1_win = Tk()
        t1_win.title('Teacher')
        t1_win.geometry('650x400+370+150')
        t1_win.resizable(False, False)
        icon7 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\office5.png', master=t1_win)
    t1_win.iconphoto(False, icon7)
        i1 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg22.png', master=t1_win)
        i2 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg4.png', master=t1_win)
        i2 = i2.subsample(2, 2)
        l1 = Label(t1_win, image=i1).place(x=-2,y=-1)
        l2 = Label(t1_win, text='Teacher', font=('Cambria', 15),
bg='black', fg='gray').place(x=300, y=1)
        l3 = Label(t1_win, text='Teacher Name : ',
font=('Cambria', 13), bg='black', fg='deep sky blue').place(x=260,
y=60)
        l4 = Label(t1_win, text=un, font=('Cambria', 13),
bg='black', fg='violet red').place(x=382, y=60)
        b1 = Button(t1_win, image=i2, text='View students mark',
compound=CENTER, command=view2, width=182, height=35,
font=('Cambria', 12), fg='white', bg='black',
activebackground='black', activeforeground='white', bd=3)
        b1.place(x=230, y=200)
        def close2():
            t1_win.withdraw()
            mainfunc()
        t1_win.protocol('WM_DELETE_WINDOW',close2)
        t1_win.mainloop()
    def view2():
        mycur.execute('select admission_no, name, class, pa1,
pa2, pa3, pa4 from class_det ')
        data = mycur.fetchall()
        data2 =[]
        t1_win.withdraw()
        view_m = Tk()
        view_m.title('View Students Marks')
        view_m.state('zoomed')
        icon8 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\office5.png', master=view_m)
    view_m.iconphoto(False, icon8)
        iv1 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\home4.png', master=view_m)
        iv1 = iv1.subsample(3, 3)
        iv2 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg26.png', master=view_m)
        l = Label(view_m, image=iv2)
        l.place(x=-3, y=-3)
        l.image = iv2
        lv1 = Label(view_m, text='View Students Mark', font=('Cambria', 15), bg='black', fg='gray').pack()
        f2 = Frame(view_m,height=20, bg='Black').pack()
        def home_v():
            view_m.withdraw()
            teacher()
        bv1 = Button(view_m, image=iv1, height=30, width=30,
bd=0, bg='black', activebackground='black', command=home_v)
        bv1.place(x=10, y=10)
        bv1.image = iv1
        s = ttk.Style(view_m)
        s.theme_use('vista')
        s.configure('Treeview', rowheight=24,
background='lightblue', foreground='black', font=('Cambria', 10))
        s.configure('Treeview.Heading', font=('Cambria', 10,
'bold'))
        col = (1, 2, 3, 4, 5, 6, 7)
        tree4 = ttk.Treeview(view_m, column=col,
show='headings', height=26)
        tree4.pack()
        sb5 = ttk.Scrollbar(view_m, orient="horizontal",
command=tree4.xview)
        sb5.pack(side='bottom', fill='x')
        tree4.configure(xscrollcommand=sb5.set)
        tree4.column(1, anchor=CENTER, stretch=NO,
width=100)
```

```
        tree4.column(2, anchor=CENTER, stretch=NO,
width=150)
        tree4.column(3, anchor=CENTER, stretch=NO, width=80)
        tree4.column(4, anchor=CENTER, stretch=NO,
width=310)
        tree4.column(5, anchor=CENTER, stretch=NO,
width=310)
        tree4.column(6, anchor=CENTER, stretch=NO,
width=310)
        tree4.column(7, anchor=CENTER, stretch=NO,
width=310)
        tree4.heading(1, text='Admission.no')
        tree4.heading(2, text='Name')
        tree4.heading(3, text='Class')
        tree4.heading(4, text='PA1 Mark')
        tree4.heading(5, text='PA2 Mark')
        tree4.heading(6, text='PA3 Mark')
        tree4.heading(7, text='PA4 Mark')
        for j in range(len(data)):
            for i in data[j]:
                data2.append(i)
            tree4.insert('', END, values=data2)
            data2.clear()
        def close3():
            view_m.withdraw()
            non_classteacher()
        view_m.protocol('WM_DELETE_WINDOW',close3)

    if Class in ['I','II','III','IV','V','VI','VII','VIII','IX','X','XI','XII']:
        teacher()

    else:
        non_classteacher()
```

Office Module:

```
from tkinter import *
from tkinter import messagebox, ttk
from datetime import date
from tkcalendar import DateEntry
import pymysql as mc
from login import *


a = b = m = n = x = y = z = q = h = k = 0
mycon = mc.connect(host='127.0.0.1', user='root', passwd='1822',
database='school')
mycur = mycon.cursor()
fee_1st = fee_2nd = gen = ''
def office_func():
    global owin
    owin = Tk()
    owin.title('Office')
    owin.geometry('650x400+370+150')
    owin.resizable(False, False)
    icon13 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\office5.png', master=owin)

    owin.iconphoto(False, icon13)

    image1 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg22.png', master=owin)

    image2 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\student3.png', master=owin)

    image2 = image2.subsample(3, 3)

    image3 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\teacher2.png', master=owin)

    image3 = image3.subsample(3, 3)

    image4 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg20.png', master=owin)

    image4 = image4.subsample(1, 1)

    label1 = Label(owin, image=image1).pack()

    l = Label(owin, text='    Office    ', font=('Cambria', 15), bg='black', fg='gray').place(x=275, y=1)

    label2 = Label(owin, image=image2, height=200, width=187, bg='black').place(x=50, y=52)

    label3 = Label(owin, image=image3, height=200, width=187, bg='black').place(x=400, y=52)
    button1 = Button(owin, image=image4, text='Student Customization', compound=CENTER, command=stu_cust, width=182, height=35, font=('Cambria', 12), fg='white', bd=3)
    button1.place(x=50, y=285)
    button2 = Button(owin, image=image4, text='Teacher Customization', compound=CENTER, command=teach_cust, width=182, height=35, font=('Cambria', 12), fg='white', bd=3)
    button2.place(x=400, y=285)
    def close():
        owin.withdraw()
        mainfunc()
    owin.protocol('WM_DELETE_WINDOW',close)
    owin.mainloop()
def stu_cust():
    global swin, tree1, data, button4, button5, button6, button7, s
    owin.withdraw()
    mycur.execute('select * from s_cust1')
    data = mycur.fetchall()
    mycur.execute('select father_name, father_number, mother_name, mother_number, guardian_name, guardian_number, annual_income, 1st_term_fee, 2nd_term_fee from s_cust2')
    data_stu = mycur.fetchall()
    data = list(data)
    for i in range(len(data)):
        for j in range(len(data_stu)):
            data[i] = list(data[i])+list(data_stu[i])
            break
    data2 = []
    swin = Tk()
    swin.state('zoomed')
    swin.title('Student Customization')
    icon14 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\office5.png', master=swin)
    swin.iconphoto(False, icon14)
    image4 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg3.png', master=swin)
    image5 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\home4.png', master=swin)
    image5 = image5.subsample(3, 3)
    label4 = Label(swin, image=image4).place(x=-3, y=-3)
    image11 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg20.png', master=swin)
    image11 = image11.subsample(1, 1)
    button3 = Button(swin, image=image5, height=30, width=30, bd=0, bg='black', activebackground='black', command=home)

    button3.place(x=10, y=10)

    frame1 = Frame(swin, height=300, width=1200)

    frame1.pack()

    image6 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg4.png', master=frame1)

    label5 = Label(frame1, image=image6).place(x=-5, y=-5)

    label0 = Label(swin, text='Student Customization', font=('Cambria', 20), bg='black', fg='gray').place(x=550,y=0)

    button4 = Button(frame1, text='ADD', image=image11, height=32, width=160, compound=CENTER, fg='white', command=add, font=('Cambria', 15, 'bold'))

    button4.place(x=100, y=90)

    button5 = Button(frame1, text='UPDATE', image=image11, height=32, width=160, compound=CENTER, fg='white', command=update, font=('Cambria', 15, 'bold'))

    button5.place(x=900, y=90)

    button6 = Button(frame1, text='REMOVE', image=image11, height=32, width=160, compound=CENTER, fg='white', command=remove, font=('Cambria', 15, 'bold'))

    button6.place(x=280, y=200)
    button7 = Button(frame1, text='SEARCH', image=image11, height=32, width=160, compound=CENTER, fg='white', command=search, font=('Cambria', 15, 'bold'))

    button7.place(x=720, y=200)

    s = ttk.Style(swin)

    s.theme_use('vista')

    s.configure('Treeview', rowheight=28, background='lightblue', foreground='black', font=('Cambria', 10))

    s.configure('Treeview.Heading', font=('Cambria', 10, 'bold'))

    sb2 = ttk.Scrollbar(swin, orient="vertical")

    sb2.pack(side='right', fill='y')

    col = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17)

    tree1 = ttk.Treeview(swin, column=col, show='headings', height=13)

    tree1.pack()

    sb1 = ttk.Scrollbar(swin, orient="horizontal", command=tree1.xview)

    sb1.pack(side='bottom', fill='x')

    tree1.configure(xscrollcommand=sb1.set)

    tree1.configure(yscrollcommand=sb2.set)

    sb2.configure(command=tree1.yview)
```

```
tree1.column(1, anchor=CENTER, stretch=NO, width=100)

tree1.column(2, anchor=CENTER, stretch=NO, width=120)

tree1.column(3, anchor=CENTER, stretch=NO, width=150)

tree1.column(4, anchor=CENTER, stretch=NO, width=100)

tree1.column(5, anchor=CENTER, stretch=NO, width=120)

tree1.column(6, anchor=CENTER, stretch=NO, width=120)

tree1.column(7, anchor=CENTER, stretch=NO, width=100)

tree1.column(8, anchor=CENTER, stretch=NO, width=120)

tree1.column(9, anchor=CENTER, stretch=NO, width=150)

tree1.column(10, anchor=CENTER, stretch=NO, width=150)

tree1.column(11, anchor=CENTER, stretch=NO, width=150)

tree1.column(12, anchor=CENTER, stretch=NO, width=150)

tree1.column(13, anchor=CENTER, stretch=NO, width=150)

tree1.column(14, anchor=CENTER, stretch=NO, width=150)

tree1.column(15, anchor=CENTER, stretch=NO, width=120)

tree1.column(16, anchor=CENTER, stretch=NO, width=150)

tree1.column(17, anchor=CENTER, stretch=NO, width=150)

tree1.heading(1, text='Admission.no')

tree1.heading(2, text='Admission date')

tree1.heading(3, text='Name')
```

```
tree1.heading(4, text='Class')
tree1.heading(5, text='Date of the Birth')
tree1.heading(6, text='Gender')
tree1.heading(7, text='Blood group')
tree1.heading(8, text='Mother tongue')
tree1.heading(9, text='Father name')
tree1.heading(10, text='Father phone number')
tree1.heading(11, text='Mother name')
tree1.heading(12, text='Mother phone number')
tree1.heading(13, text='Guardian name')
tree1.heading(14, text='Guardian phone number')
tree1.heading(15, text='Annual income')
tree1.heading(16, text='First Term Fees')
tree1.heading(17, text='Second Term Fees')
for j in range(len(data)):
    for i in data[j]:
        data2.append(i)
    tree1.insert('', END, values=data2)
    data2.clear()
    def close():
        swin.withdraw()
        office_func()
    swin.protocol('WM_DELETE_WINDOW',close)
    swin.mainloop()
def home():
    swin.withdraw()
    office_func()
def add():
    def save():
        in_data = list()
        in_values = list()
        in_values1 = list()
        mark_data = list()
        mark_values = list()
        in_data.append(int(a))
        in_values.append('admission_no')
        in_values1.append('admission_no')
        mark_values.append('admission_no')
        mark_data.append(a)
        in_data.append(d_entry1.get().strip())
        in_values.append('admission_date')
        image9 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d1.png', master=add_win)
        image10 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d2.png', master=add_win)
        image11 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d3.png', master=add_win)
        image12 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d4.png', master=add_win)
        image13 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d5.png', master=add_win)
        image14 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d6.png', master=add_win)
        image15 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d7.png', master=add_win)
        image16 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d8.png', master=add_win)
        image17 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d9.png', master=add_win)
        image18 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d10.png', master=add_win)
        if entry2.get().strip() != '':
            in_data.append(entry2.get().strip())
            in_values.append('name')
            mark_values.append('name')
            mark_data.append(entry2.get().strip())
        else:
            label24 = Label(add_win, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image9, compound=CENTER)
            label24.image = image9
            label24.place(x=555, y=187)
            add_win.after(2500, label24.destroy)
        if combo3.get().strip() != '':
            in_data.append(combo3.get().strip())
            in_values.append('class')
            mark_data.append(combo3.get().strip())
            mark_values.append('class')
        else:
            label25 = Label(add_win, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image9, compound=CENTER)
            label25.image = image9
            label25.place(x=570, y=251)
            add_win.after(2500, label25.destroy)
        in_data.append(d_entry2.get().strip())
        in_values.append('dob')
        in_data.append(gen)
        in_values.append('gender')
        if combo1.get().strip() != '':
            in_data.append(combo1.get().strip())
            in_values.append('blood_group')
        else:
            label26 = Label(add_win, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image10, compound=CENTER)
            label26.image = image10
            label26.place(x=570, y=446)
            add_win.after(2500, label26.destroy)
        if entry5.get().strip() != '':
            in_data.append(entry5.get().strip())
            in_values.append('mother_tongue')
        else:
            pass
            label27 = Label(add_win, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image11, compound=CENTER)

            label27.image = image11

            label27.place(x=553, y=512)

            add_win.after(2500, label27.destroy)

        if entry6.get() == '' and entry7.get() == '' and entry8.get() == '' and entry9.get() == '' and entry10.get() == '' and entry11.get() == '':

            label28 = Label(add_win, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image13, compound=CENTER)

            label28.image = image13

            label28.place(x=553, y=575)

            add_win.after(2500, label28.destroy)

            label29 = Label(add_win, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image14, compound=CENTER)

            label29.image = image14

            label29.place(x=553, y=640)

            add_win.after(2500, label29.destroy)

            label30 = Label(add_win, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image15, compound=CENTER)
```

```
            label30.image = image15

            label30.place(x=1315, y=60)

            add_win.after(2500, label30.destroy)

            label31 = Label(add_win, text='●', font=('Cambria', 14, 'bold'),
fg='red', height=20, width=10, image=image16,
compound=CENTER)

            label31.image = image16

            label31.place(x=1315, y=125)

            add_win.after(2500, label31.destroy)

            label32 = Label(add_win, text='●', font=('Cambria', 14, 'bold'),
fg='red', height=20, width=10, image=image17,
compound=CENTER)

            label32.image = image17

            label32.place(x=1315, y=190)

            add_win.after(2500, label32.destroy)

            label33 = Label(add_win, text='●', font=('Cambria', 14, 'bold'),
fg='red', height=20, width=10, image=image18,
compound=CENTER)

            label33.image = image18

            label33.place(x=1315, y=255)

            add_win.after(2500, label33.destroy)

        else:
```

```
        if entry6.get().strip() != '' and entry7.get().strip() != '':
            in_data.append(entry6.get().strip())
            in_values1.append('father_name')
            if entry7.get().strip().isdigit() and len(entry7.get().strip())
== 10:
                in_data.append(int(entry7.get().strip()))
                in_values1.append('father_number')
            else:
                label36 = Label(add_win, text='Invalid Phone number ! ',
font=('Cambria', 12, 'bold'), fg='red', bg='black')
                label36.place(x=330, y=675)
                add_win.after(2500, label36.destroy)
        if entry6.get().strip() != '' and entry7.get().strip() == '':
            in_data.append(entry6.get().strip())
            in_values1.append('father_name')
            label29 = Label(add_win, text='●', font=('Cambria', 14,
'bold'), fg='red', height=20, width=10, image=image14,
compound=CENTER)
            label29.image = image14
            label29.place(x=553, y=640)
            add_win.after(2500, label29.destroy)
        if entry6.get().strip() == '' and entry7.get().strip() != '':
            if entry7.get().strip().isdigit() and len(entry7.get().strip())
== 10:
                label28 = Label(add_win, text='●', font=('Cambria', 14,
'bold'), fg='red', height=20, width=10, image=image13,
compound=CENTER)
                label28.image = image13
                label28.place(x=553, y=575)
                add_win.after(2500, label28.destroy)
            else:
                label28 = Label(add_win, text='●', font=('Cambria', 14,
'bold'), fg='red', height=20, width=10, image=image13,
compound=CENTER)
                label28.image = image13
                label28.place(x=553, y=575)
                add_win.after(2500, label28.destroy)
                label36 = Label(add_win, text='Invalid Phone number ! ',
font=('Cambria', 12, 'bold'), fg='red', bg='black')
                label36.place(x=330, y=675)
                add_win.after(2500, label36.destroy)
        if entry8.get().strip() != '' and entry9.get().strip() != '':
            in_data.append(entry8.get().strip())
            in_values1.append('mother_name')
            if entry9.get().strip().isdigit() and len(entry9.get().strip()) == 10:
                in_data.append(int(entry9.get().strip()))
                in_values1.append('mother_number')
            else:
                label36 = Label(add_win, text='Invalid Phone number ! ', font=('Cambria', 12, 'bold'), fg='red', bg='black')
                label36.place(x=1090, y=160)
                add_win.after(2500, label36.destroy)
        if entry8.get().strip() != '' and entry9.get().strip() == '':
            in_data.append(entry8.get().strip())
            in_values1.append('mother_name')
            label31 = Label(add_win, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image16, compound=CENTER)
            label31.image = image16
            label31.place(x=1315, y=125)
            add_win.after(2500, label31.destroy)

        if entry8.get().strip() == '' and entry9.get().strip() != '':
            if entry9.get().strip().isdigit() and len(entry9.get().strip()) == 10:
                in_data.append(int(entry9.get().strip()))
                in_values1.append('mother_number')
                label30 = Label(add_win, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image15, compound=CENTER)
                label30.image = image15
                label30.place(x=1315, y=60)
                add_win.after(2500, label30.destroy)
            else:
                label30 = Label(add_win, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image15, compound=CENTER)
                label30.image = image15
                label30.place(x=1315, y=60)
                add_win.after(2500, label30.destroy)
                label39 = Label(add_win, text='Invalid Phone number ! ', font=('Cambria', 12, 'bold'), fg='red', bg='black')
                label39.place(x=1090, y=160)
                add_win.after(2500, label39.destroy)
        if entry10.get().strip() != '' and entry11.get().strip() != '':
            in_data.append(entry10.get().strip())

            in_values1.append('guardian_name')

            if entry11.get().strip().isdigit() and
len(entry11.get().strip()) == 10:

                in_data.append(int(entry11.get().strip()))

                in_values1.append('guardian_number')

            else:

                label36 = Label(add_win, text='Invalid Phone number ! ',
font=('Cambria', 12, 'bold'), fg='red', bg='black')

                label36.place(x=1090, y=290)

                add_win.after(2500, label36.destroy)

        if entry10.get().strip() != '' and entry11.get().strip() == '':

            in_data.append(entry10.get().strip())

            in_values1.append('guardian_name')

            label33 = Label(add_win, text='●', font=('Cambria', 14,
'bold'), fg='red', height=20, width=10, image=image18,
compound=CENTER)

            label33.image = image18

            label33.place(x=1315, y=255)

            add_win.after(2500, label33.destroy)
        if entry10.get().strip() == '' and entry11.get().strip() != '':
            if entry11.get().strip().isdigit() and len(entry11.get().strip()) == 10:
                in_data.append(entry11.get().strip())
                in_values1.append('guardian_number')
                label32 = Label(add_win, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image17, compound=CENTER)
                label32.image = image17
                label32.place(x=1315, y=190)
                add_win.after(2500, label32.destroy)
            else:
                label32 = Label(add_win, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image17, compound=CENTER)
                label32.image = image17
                label32.place(x=1315, y=190)
                add_win.after(2500, label32.destroy)
                label42 = Label(add_win, text='Invalid Phone number ! ', font=('Cambria', 12, 'bold'), fg='red', bg='black')
                label42.place(x=1090, y=290)
                add_win.after(2500, label42.destroy)
```

```
    if entry12.get().strip() != '':
        if entry12.get().strip().isalpha() == False :
            in_data.append(entry12.get())
            in_values1.append('annual_income')
        else:
            label34 = Label(add_win, text='Invalid income ! ',
font=('Cambria', 12, 'bold'), fg='red', bg='black')
            label34.place(x=1120, y=355)
            add_win.after(2500, label34.destroy)
    else:
        label35 = Label(add_win, text='●', font=('Cambria', 14, 'bold'),
fg='red', height=20, width=10, image=image12,
compound=CENTER)
        label35.image = image12
        label35.place(x=1315, y=320)
        add_win.after(2500, label35.destroy)
    in_data.append(fee_1st)
    in_values1.append('1st_term_fee')
    in_data.append(fee_2nd)
    in_values1.append('2nd_term_fee')
    mark_values.append('subject')
    if len(mark_data) == 3 :
        if mark_data[2] in ['XI','XII']:
            mark_data.append('eng,mat,chem,phy,cs')
        else:
            mark_data.append('eng,mat,sci,sco,IIlang')
    in_data1 = in_data[:8]
    in_data2 = [in_data[0]]+in_data[8:]
    if len(in_values) != 0 and len(in_data) != 0 and len(in_values1) !=
0 and len(in_data2) != 0 and len(mark_values) != 0 and
len(mark_data) != 0:
        if len(in_values) == len(in_data1) and len(in_values1) ==
len(in_data2) and len(mark_values) == len(mark_data) :
            code1 = 'insert into s_cust1
'+str(tuple(in_values)).replace("'",'')+' values '+str(tuple(in_data1))
            code2 = 'insert into s_cust2
'+str(tuple(in_values1)).replace("'",'')+' values
'+str(tuple(in_data2))
            code3 = 'insert into class_det
'+str(tuple(mark_values)).replace("'",'')+' values
'+str(tuple(mark_data))
            mycur.execute(code1)
            mycur.execute(code2)
```

```
                mycur.execute(code3)

                mycur.execute('commit')

                add_win.withdraw()

                if messagebox.showinfo('Info','Student details successfully saved .'):

                    stu_cust()

    def reset():

        d_entry1.delete(0, END)

        d_entry1.insert(0, date1)

        entry2.delete(0, END)

        entry2.insert(0, '')

        combo3.delete(0, END)

        combo3.insert(0, '')

        d_entry2.delete(0, END)

        dob2 = str(date.today())[8:] + '/' + str(date.today())[5:7] + '/' + str(int(str(date.today())[:4]) - 4)

        d_entry2.insert(0, dob2)

        checkbutton1.deselect()

        checkbutton2.deselect()

        checkbutton3.deselect()

        combo1.delete(0, END)
```

```
combo1.insert(0, '')
entry5.delete(0, END)
entry5.insert(0, '')
entry6.delete(0, END)
entry6.insert(0, '')
entry7.delete(0, END)
entry7.insert(0, '')
entry8.delete(0, END)
entry8.insert(0, '')
entry9.delete(0, END)
entry9.insert(0, '')
entry10.delete(0, END)
entry10.insert(0, '')
entry11.delete(0, END)
entry11.insert(0, '')
entry12.delete(0, END)
entry12.insert(0, '')
if x == 1:
    if fee_1st == 'Partially Paid':
        back1()
        if y == 1:
            if fee_2nd == 'Partially Paid':
                back2()
        checkbutton4.deselect()
        checkbutton5.deselect()
        checkbutton6.deselect()
        checkbutton7.deselect()
        checkbutton8.deselect()
        checkbutton9.deselect()
    def cancel():
        add_win.withdraw()
        if messagebox.askyesno('', 'Are you sure to Exit ?'):
            stu_cust()
    swin.withdraw()
    add_win = Tk()
    add_win.title('Append students data')
    add_win.state('zoomed')
    icon15 = PhotoImage(file=r'C:\Users\User\Desktop\New
folder\office5.png', master=add_win)
    add_win.iconphoto(False, icon15)
    image7 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg18.png', master=add_win)
    label23 = Label(add_win, image=image7).place(x=-2, y=-2)
    a = str(int(data[-1][0])+1)
    label3 = Label(add_win, text='APPEND DATA', font=('Cambria', 20), bg='black', fg='gray').pack()
    label4 = Label(add_win, text='Admission Number : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=50, y=60)
    label5 = Label(add_win, text='  '+a+'  ', font=('Cambria', 15, 'bold'), bg='black', fg='cyan').place(x=300, y=60)
    label6 = Label(add_win, text='Admission Date : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=50, y=120)
    d_entry1 = DateEntry(add_win, font=('Cambria', 15, 'bold'), width=20, selectmode='day', maxdate=date.today(), background='black', foreground='white', date_pattern='dd/mm/y')
    d_entry1.place(x=300, y=121)
    date1 = d_entry1.get()
    dob = date1.replace(date1[6:], str(int(date1[6:])-4))
    dob = date(int(dob[6:]), int(dob[3:5]), int(dob[:2]))
    d_entry1.focus()
    label7 = Label(add_win, text='Name of the Student : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=50, y=185)
    entry2 = ttk.Entry(add_win, font=('Cambria', 15, 'bold'))
    entry2.place(x=300, y=186)
    label8 = Label(add_win, text='Class : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=50, y=250)
    list3 = ['I', 'II', 'III', 'IV', 'V', 'VI', 'VII', 'VIII', 'IX', 'X', 'XI', 'XII']
    combo3 = ttk.Combobox(add_win, value=list3, font=('Cambria', 15, 'bold'))
    combo3.place(x=300, y=251)
    label9 = Label(add_win, text='Date of Birth : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=50, y=315)
    d_entry2 = DateEntry(add_win, font=('Cambria', 15, 'bold'), width=20, selectmode='day', maxdate=dob, background='black', foreground='white', date_pattern='dd/mm/y')
    d_entry2.place(x=300, y=316)
    label10 = Label(add_win, text='Gender : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=50, y=380)
    def male():
        global gen
        gen = 'male'
    def female():
        global gen
        gen = 'female'
    def others():
        global gen
        gen = 'others'
    checkbutton1 = Checkbutton(add_win, text='Male', bg='lightblue',
activebackground='lightblue', height=1, variable=1, onvalue=1,
offvalue=0, font=('Cambria', 12, 'bold'), command=male)
    checkbutton1.place(x=300, y=380)
    checkbutton2 = Checkbutton(add_win, text='Female',
bg='lightblue', activebackground='lightblue', height=1, variable=1,
onvalue=2, offvalue=0, font=('Cambria', 12, 'bold'),
command=female)
    checkbutton2.place(x=400, y=380)
    checkbutton3 = Checkbutton(add_win, text='Others',
bg='lightblue', activebackground='lightblue', height=1, variable=1,
onvalue=3, offvalue=0, font=('Cambria', 12, 'bold'),
command=others)
    checkbutton3.place(x=525, y=380)
    label11 = Label(add_win, text='Blood Group : ', font=('Cambria',
15, 'bold'), bg='black', fg='white').place(x=50, y=445)
    list1 = ['O+ve', 'O-ve', 'A+ve', 'A-ve', 'B+ve', 'B-ve', 'AB+ve', 'AB-ve',
'A1+ve', 'Hh']
    combo1 = ttk.Combobox(add_win, value=list1, font=('Cambria',
15, 'bold'))
```

```
    combo1.place(x=300, y=446)
    label12 = Label(add_win, text='Mother Tongue : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=50, y=510)
    entry5 = ttk.Entry(add_win, font=('Cambria', 15, 'bold'))
    entry5.place(x=300, y=511)
    label13 = Label(add_win, text="Father's Name : ", font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=50, y=575)
    entry6 = ttk.Entry(add_win, font=('Cambria', 15, 'bold'))
    entry6.place(x=300, y=576)
    label14 = Label(add_win, text="Father's Phone Number : ", font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=50, y=640)
    entry7 = ttk.Entry(add_win, font=('Cambria', 15, 'bold'))
    entry7.place(x=300, y=641)
    label15 = Label(add_win, text="Mother's Name : ", font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=800, y=60)
    entry8 = ttk.Entry(add_win, font=('Cambria', 15, 'bold'))
    entry8.place(x=1060, y=61)
    label16 = Label(add_win, text="Mother's Phone Number : ", font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=800, y=125)
    entry9 = ttk.Entry(add_win, font=('Cambria', 15, 'bold'))
    entry9.place(x=1060, y=126)
    label17 = Label(add_win, text="Guardian Name : ", font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=800, y=190)
    entry10 = ttk.Entry(add_win, font=('Cambria', 15, 'bold'))
    entry10.place(x=1060, y=191)

    label18 = Label(add_win, text="Guardian Phone Number : ", font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=800, y=255)
    entry11 = ttk.Entry(add_win, font=('Cambria', 15, 'bold'))
    entry11.place(x=1060, y=256)
    label19 = Label(add_win, text="Annual Income : ", font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=800, y=320)
    entry12 = ttk.Entry(add_win, font=('Cambria', 15, 'bold'))
    entry12.place(x=1060, y=321)
    def paid1():
        global fee_1st
        fee_1st = 'Paid'
    def not_paid1():
        global fee_1st
        fee_1st = 'Not_Paid'
    def partially_paid1():
        global fee_1st, back1, x
        def back1():
            button21.destroy()
            label21.destroy()
            entry13.destroy()
            checkbutton6.deselect()
            checkbutton4['state'] = ACTIVE
            checkbutton5['state'] = ACTIVE
            checkbutton6['state'] = ACTIVE
        x = 1
        fee_1st = 'Partially Paid'
        checkbutton4['state'] = DISABLED
        checkbutton5['state'] = DISABLED
        checkbutton6['state'] = DISABLED
        image8 = PhotoImage(file=r'C:\Users\User\Desktop\New
folder\back1.png', master=add_win)
```

```
        button21 = Button(add_win, width=20, height=20, image=image8, bd=0, command=back1, bg='black', activebackground='lightblue')

        button21.place(x=750, y=463)

        button21.image = image8

        label21 = Label(add_win, text='Balance Amount : ', font=('Cambria', 15, 'bold'), bg='black', fg='white')

        label21.place(x=800, y=460)

        entry13 = ttk.Entry(add_win, font=('Cambria', 15, 'bold'))

        entry13.place(x=1060, y=461)

        entry13.focus()

    label20 = Label(add_win, text='First Term Fees : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=800, y=385)

    checkbutton4 = Checkbutton(add_win, text='Paid', bg='lightblue', activebackground='lightblue', height=1, variable=2, onvalue=1, offvalue=0, font=('Cambria', 12, 'bold'), command=paid1)

    checkbutton4.place(x=1060, y=385)

    checkbutton5 = Checkbutton(add_win, text='Not Paid', bg='lightblue', activebackground='lightblue', height=1, variable=2, onvalue=2, offvalue=0, font=('Cambria', 12, 'bold'), command=not_paid1)

    checkbutton5.place(x=1200, y=385)
    checkbutton6 = Checkbutton(add_win, text='Partially Paid',
bg='lightblue', activebackground='lightblue', height=1, variable=2,
onvalue=3, offvalue=0, font=('Cambria', 12, 'bold'),
command=partially_paid1)

    checkbutton6.place(x=1100, y=422)

    def paid2():

        global fee_2nd

        fee_2nd = 'Paid'

    def not_paid2():

        global fee_2nd

        fee_2nd = 'Not Paid'

    def partially_paid2():

        global fee_2nd, back2, y

        def back2():

            button20.destroy()

            label23.destroy()

            entry14.destroy()

            checkbutton9.deselect()

            checkbutton7['state'] = ACTIVE

            checkbutton8['state'] = ACTIVE

            checkbutton9['state'] = ACTIVE
        y = 1

        fee_2nd = 'Partially Paid'

        checkbutton7['state'] = DISABLED

        checkbutton8['state'] = DISABLED

        checkbutton9['state'] = DISABLED

        image7 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\back1.png', master=add_win)

        button20 = Button(add_win, width=20, height=20, image=image7, bd=0, command=back2, bg='black', activebackground='lightblue')

        button20.place(x=750, y=603)

        button20.image = image7

        label23 = Label(add_win, text='Balance Amount : ', font=('Cambria', 15, 'bold'), bg='black', fg='white')

        label23.place(x=800, y=600)

        entry14 = ttk.Entry(add_win, font=('Cambria', 15, 'bold'))

        entry14.place(x=1060, y=601)

        entry14.focus()

    label22 = Label(add_win, text='Second Term Fees : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=800, y=525)
    checkbutton7 = Checkbutton(add_win, text='Paid', bg='lightblue', activebackground='lightblue', height=1, variable=3, onvalue=1, offvalue=0, font=('Cambria', 12, 'bold'), command=paid2)

    checkbutton7.place(x=1060, y=525)

    checkbutton8 = Checkbutton(add_win, text='Not Paid', bg='lightblue', activebackground='lightblue', height=1, variable=3, onvalue=2, offvalue=0, font=('Cambria', 12, 'bold'), command=not_paid2)

    checkbutton8.place(x=1200, y=525)

    checkbutton9 = Checkbutton(add_win, text='Partially Paid', bg='lightblue', activebackground='lightblue', height=1, variable=3, onvalue=3, offvalue=0, font=('Cambria', 12, 'bold'), command=partially_paid2)

    checkbutton9.place(x=1100, y=562)

    image9 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg20.png', master=add_win)

    image9 = image9.subsample(2, 2)

    button13 = Button(add_win, text='Save', font=('Cambria', 15, 'bold'), image=image9, fg='white', height=30, width=120, command=save, compound=CENTER)

    button13.place(x=800, y=660)

    button14 = Button(add_win, text='Reset', font=('Cambria', 15, 'bold'), image=image9, fg='white', height=30, width=120, command=reset, compound=CENTER)
    button14.place(x=1000, y=660)

    button15 = Button(add_win, text='Cancel', font=('Cambria', 15,
'bold'), image=image9, fg='white', height=30, width=120,
command=cancel, compound=CENTER)

    button15.place(x=1200, y=660)

    add_win.mainloop()

def remove():

    for i in [button4, button5, button6, button7]:

        i.destroy()

    swin.title('Remove students data')

    def back1r():

        swin.withdraw()

        stu_cust()

    def remove_single():

        global label9, label10, entry1, entry2

        button13.destroy()

        button9.destroy()

        button10.destroy()

        lable9 = Label(swin, text='Admission Number : ',
font=('Cambria', 15), bg='black', fg='gray')

        lable9.place(x=150, y=125)
        lable10 = Label(swin, text='Name of the Student : ',
font=('Cambria', 15), bg='black', fg='gray')
        lable10.place(x=760, y=125)
        entry1 = ttk.Entry(swin, font=('Cambria', 15, 'bold'))
        entry1.place(x=350, y=125)
        entry2 = ttk.Entry(swin, font=('Cambria', 15, 'bold'))
        entry2.place(x=970, y=125)
        def show_single():
            global list_r, m, button12
            list_r =[]
            if entry1.get() != '' and entry2.get() == '':
                m = 2
                for i in tree1.get_children():
                    tree1.delete(i)
                mycur.execute('select * from s_cust1')
                data = mycur.fetchall()
                mycur.execute('select father_name, father_number,
mother_name, mother_number, guardian_name, guardian_number,
annual_income, 1st_term_fee, 2nd_term_fee from s_cust2')
                data_stu = mycur.fetchall()
                data = list(data)
```

```
                for i in range(len(data)):
                    for j in range(len(data_stu)):
                        data[i] = list(data[i])+list(data_stu[i])
                        break
                flag1 = 0
                for i in data:
                    if str(entry1.get().strip()) == str(i[0]):
                        button11.destroy()
                        tree1.insert('', END, values=i)
                        list_r.append(i)
                        def delete_single():
                            swin.withdraw()
                            m1 = messagebox.askyesno('', 'Are you sure to Delete ?')
                            if m1 == True:
                                for i in list_r:
                                    code5 = 'delete from s_cust1 where admission_no= '+str(i[0])
                                    code6 = 'delete from s_cust2 where admission_no= '+str(i[0])
                                    code7 = 'delete from class_det where admission_no= '+str(i[0])
                                    mycur.execute(code5)
                                    mycur.execute(code6)
                                    mycur.execute(code7)
                                    mycon.commit()
                                    if messagebox.showinfo('Info','Student details successfully removed .'):
                                        stu_cust()
                            else:
                                stu_cust()
                        button12 = Button(swin, text='Delete', font=('Cambria', 15, 'bold'), fg='white', image=image10, height=25, width=100, compound=CENTER, command=delete_single)
                        button12.image = image10
                        button12.place(x=623, y=225)
                        entry1['state'] = DISABLED
                        entry2['state'] = DISABLED
                    else:
                        flag1+=1
                if flag1 == len(data):
                    m = 1
                    label12 = Label(swin, text='Not Found', font=('Cambria',
12, 'bold'), fg='red', bg='black')
                    label12.place(x=637, y=185)
                    swin.after(2500, label12.destroy)
            elif entry1.get() == '' and entry2.get() != '':
                m = 2
                for i in tree1.get_children():
                    tree1.delete(i)
                mycur.execute('select * from s_cust1')
                data = mycur.fetchall()
                mycur.execute('select father_name, father_number,
mother_name, mother_number, guardian_name, guardian_number,
annual_income, 1st_term_fee, 2nd_term_fee from s_cust2')
                data_stu = mycur.fetchall()
                data = list(data)
                for i in range(len(data)):
                    for j in range(len(data_stu)):
                        data[i] = list(data[i])+list(data_stu[i])
                        break
                flag2 = 0
```

```
for i in data:
    if entry2.get() == i[2]:
        button11.destroy()
        tree1.insert('', END, values=i)
        list_r.append(i)
        if len(list_r)>1:
            def sel(event):
                for j in tree1.selection():
                    item = tree1.item(j)
                    record = item['values']
                tree1.insert('', END, values=record)
            tree1.bind('<Double-1>', sel)
        def delete_single():
            swin.withdraw()
            m1 = messagebox.askyesno('', 'Are you sure to Delete ?')
            if m1 == True:
                for i in list_r:
                    code5 = 'delete from s_cust1 where admission_no= '+str(i[0])
                    code6 = 'delete from s_cust2 where
admission_no= '+str(i[0])

                    code7 = 'delete from class_det where
admission_no= '+str(i[0])

                    mycur.execute(code5)

                    mycur.execute(code6)

                    mycur.execute(code7)

                    mycon.commit()

                    if messagebox.showinfo('Info','Student details
successfully removed .'):

                  stu_cust()

            else:

                stu_cust()

        button12 = Button(swin, text='Delete', font=('Cambria',
15, 'bold'), fg='white', image=image10, height=25, width=100,
compound=CENTER, command=delete_single)

        button12.place(x=623, y=225)

        button12.image = image10

        entry1['state'] = DISABLED

        entry2['state'] = DISABLED

    else:
                flag2 += 1
            if flag2 == len(data):
                m = 1
                label12 = Label(swin, text='Not Found', font=('Cambria',
12, 'bold'), fg='red', bg='black')
                label12.place(x=637, y=185)
                swin.after(2500, label12.destroy)
        elif entry1.get() != '' and entry2.get() != '':
            m = 2
            for i in tree1.get_children():
                tree1.delete(i)
            mycur.execute('select * from s_cust1')
            data = mycur.fetchall()
            mycur.execute('select father_name, father_number,
mother_name, mother_number, guardian_name, guardian_number,
annual_income, 1st_term_fee, 2nd_term_fee from s_cust2')
            data_stu = mycur.fetchall()
            data = list(data)
            for i in range(len(data)):
                for j in range(len(data_stu)):
                    data[i] = list(data[i])+list(data_stu[i])
            break
        flag3 = 0
        for i in data:
          if str(entry1.get()) == str(i[0]) and entry2.get().lower()
== i[2].lower():
            button11.destroy()
            tree1.insert('', END, values=i)
            list_r.append(i)
            def delete_single():
              swin.withdraw()
              m1 = messagebox.askyesno('', 'Are you sure to Delete
?')
              if m1 == True:
                for i in list_r:
                  code5 = 'delete from s_cust1 where
admission_no= '+str(i[0])
                  code6 = 'delete from s_cust2 where
admission_no= '+str(i[0])
                  code7 = 'delete from class_det where
admission_no= '+str(i[0])
                  mycur.execute(code5)
```

```
                        mycur.execute(code6)
                        mycur.execute(code7)
                        mycon.commit()
                        if messagebox.showinfo('Info','Student details successfully removed .'):
                            stu_cust()
                else:
                    stu_cust()
            button12 = Button(swin, text='Delete', font=('Cambria', 15, 'bold'), fg='white', image=image10, height=25, width=100, compound=CENTER, command=delete_single)
            button12.place(x=623, y=225)
            button12.image = image10
            entry1['state'] = DISABLED
            entry2['state'] = DISABLED
        else:
            flag3+=1
    if flag3 == len(data):
        m = 1
        label12 = Label(swin, text='Not Found', font=('Cambria', 12, 'bold'), fg='red', bg='black')
```

```
                    label12.place(x=637, y=185)
                    swin.after(2500, label12.destroy)
            elif entry1.get() == '' and entry2.get() == '':
                global a, b
                m = 1
                a += 1
                for i in range(1, 5):
                    if a == b or (a, b == 1, 2):
                        global label11
                        label11 = Label(swin, text='The required information is empty', font=('Cambria', 10, 'bold'), fg='red', bg='black')
                        label11.place(x=568, y=185)
                        swin.after(2500, label11.destroy)
                        break
                    else:
                        a -= 1
                        break
        def back2r():
            lable9.destroy()
            lable10.destroy()
            entry1.destroy()

            entry2.destroy()

            if m == 1 or m == 0:

                button11.destroy()

            elif m == 2:

                button12.destroy()

                button12.destroy()

                button11.destroy()

            remove()

        button11 = Button(swin, text='Show', font=('Cambria', 15, 'bold'), fg='white', image=image10, height=25, width=100, compound=CENTER, command=show_single)

        button11.place(x=623, y=225)

        button14 = Button(swin, text='Back', fg='white', image=image10, height=20, width=75, compound=CENTER, font=('Cambria', 15), command=back2r)

        button14.place(x=1180, y=265)

    def remove_multi():

        button9.destroy()

        button10.destroy()

        button13.destroy()
        def show_multi():
            global lsm, list_rm, n, button12
            list_rm = []
            f = 0
            f2 = 0
            sm_dict = {}
            lsm = []
            if combo1r.get().strip() == '' and combo2r.get().strip() == '' and combo3r.get().strip() == '' and combo4r.get().strip() == '' and combo5r.get().strip() == '':
                n = 1
                lable1md = Label(swin, text='The required information is empty', font=('Cambria', 12, 'bold'), bg='black', fg='red')
                lable1md.place(x=543, y=160)
                swin.after(2500, lable1md.destroy)
            elif combo1r.get().strip() != '' or combo2r.get().strip() != '' or combo3r.get().strip() != '' or combo4r.get().strip() != '' or combo5r.get().strip() != '':
                for i in tree1.get_children():
                    tree1.delete(i)
                get_sm = [combo1r.get().strip(), combo2r.get().strip(), combo3r.get().strip(), combo4r.get().strip(), combo5r.get().strip()]
                mycur.execute('select * from s_cust1')
                data2 = mycur.fetchall()
                mycur.execute('select father_name, father_number,
        mother_name, mother_number, guardian_name, guardian_number,
        annual_income, 1st_term_fee, 2nd_term_fee from s_cust2')
                data_stu = mycur.fetchall()
                data2 = list(data2)
                for i in range(len(data2)):
                    for j in range(len(data_stu)):
                        data2[i] = list(data2[i])+list(data_stu[i])
                        break
                mycur.execute('select class,gender,blood_group from
        s_cust1')
                data3 = mycur.fetchall()
                mycur.execute('select 1st_term_fee, 2nd_term_fee from
        s_cust2')
                data_stu1 = mycur.fetchall()
                data3 = list(data3)
                for i in range(len(data3)):
                    for j in range(len(data_stu1)):
                        data3[i] = list(data3[i])+list(data_stu1[i])
                        break
                for i in get_sm:
                    if i != '':
                        sm_dict[get_sm.index(i)] = i
                        index1 = get_sm.index(i)
                        get_sm.remove(i)
                        get_sm.insert(index1, '')
                for i in data3:
                    for j in sm_dict:
                        if str(sm_dict[j]).lower() == str(i[j]).lower():
                            f += 1
                        if f == len(sm_dict):
                            list_rm.append(data2[data3.index(i)])
                            tree1.insert('', END, values=data2[data3.index(i)])
                            lsm.append(data2[data3.index(i)])
                            n = 2
                            f2 = 1
                            def delete_multi():
                                swin.withdraw()
                            m4 = messagebox.askyesno('', 'Are you sure to 
                Delete ?')
                            if m4 == True:
                                button11.destroy()
                                for i in list_rm:
                                    code7 = 'delete from class_det where 
                admission_no= '+str(i[0])
                                    code8 = 'delete from s_cust1 where 
                admission_no=' + str(i[0])
                                    code9 = 'delete from s_cust2 where 
                admission_no=' + str(i[0])
                                    mycur.execute(code7)
                                    mycur.execute(code8)
                                    mycur.execute(code9)
                                    mycon.commit()
                                    if messagebox.showinfo('Info','Student details 
                successfully removed .'):
                                        stu_cust()
                        button12 = Button(swin, text='Delete', 
                font=('Cambria', 15, 'bold'), fg='white', image=image10, height=25, 
                width=100, compound=CENTER, command=delete_multi)
                        button12.place(x=615, y=250)
```

```
                            combo1r['state'] = DISABLED
                            combo2r['state'] = DISABLED
                            combo3r['state'] = DISABLED
                            combo4r['state'] = DISABLED
                            combo5r['state'] = DISABLED
                    f = 0
                if f != len(sm_dict) and f2 != 1:
                    n = 1
                    for k in tree1.get_children():
                        tree1.delete(k)
                    label12 = Label(swin, text='Not Found', font=('Cambria',
10, 'bold'), fg='red', bg='black')
                    label12.place(x=625, y=200)
                    swin.after(2500, label12.destroy)
        def back3r():
            combo1r.destroy()
            combo2r.destroy()
            combo3r.destroy()
            combo4r.destroy()
            combo5r.destroy()
            label1r.destroy()
            label2r.destroy()
            label3r.destroy()
            label4r.destroy()
            label5r.destroy()
            if n==2:
                button12.destroy()
                button11.destroy()
            else:
                button11.destroy()
            remove()
        label1r = Label(swin, text='Class : ', font=('Cambria', 15, 'bold'), bg='black', fg='white')
        label1r.place(x=180, y=50)
        list1r = ['I', 'II', 'III', 'IV', 'V', 'VI', 'VII', 'VIII', 'IX', 'X', 'XI', 'XII']
        combo1r = ttk.Combobox(swin, value=list1r, font=('Cambria', 15, 'bold'))
        combo1r.place(x=280, y=50)
        label2r=Label(swin, text='Gender : ', font=('Cambria', 15, 'bold'), bg='black', fg='white')
        label2r.place(x=800, y=50)
        list2r = ['Male', 'Female', 'Others']
        combo2r = ttk.Combobox(swin, value=list2r, font=('Cambria', 15, 'bold'))
        combo2r.place(x=925, y=50)
        label3r = Label(swin, text='Blood Group : ', font=('Cambria', 15, 'bold'), bg='black', fg='white')
        label3r.place(x=490, y=120)
        list3r = ['O+ve', 'O-ve', 'A+ve', 'A-ve', 'B+ve', 'B-ve', 'AB+ve', 'AB-ve', 'A1+ve', 'Hh']
        combo3r = ttk.Combobox(swin, value=list3r, font=('Cambria', 15, 'bold'))
        combo3r.place(x=660, y=120)
        label4r = Label(swin, text='Term I : ', font=('Cambria', 15, 'bold'), bg='black', fg='white')
        label4r.place(x=180, y=200)
        list4r = ['Paid', 'Partially Paid', 'Not Paid']
        combo4r = ttk.Combobox(swin, value=list4r, font=('Cambria', 15, 'bold'))
        combo4r.place(x=280, y=200)
        label5r = Label(swin, text='Term II : ', font=('Cambria', 15, 'bold'), bg='black', fg='white')
        label5r.place(x=800, y=200)
```

```
        combo5r = ttk.Combobox(swin, value=list4r, font=('Cambria', 15, 'bold'))
        combo5r.place(x=925, y=200)
        button11 = Button(swin, text='Show', font=('Cambria', 15, 'bold'), fg='white', image=image10, height=25, width=100, compound=CENTER, command=show_multi)
        button11.place(x=615, y=250)
        button15 = Button(swin, text='Back', fg='white', image=image10, height=20, width=75, compound=CENTER, font=('Cambria', 15), command=back3r)
        button15.place(x=1180, y=265)
    image10 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg20.png', master=swin)
    image10 = image10.subsample(1, 1)
    button9 = Button(swin, text='Remove Single Records', command=remove_single, fg='white', image=image10, height=70, width=290, compound=CENTER, font=('Cambria', 15))
    button9.place(x=305, y=120)
    button10 = Button(swin, text='Remove Multi Records', command=remove_multi, fg='white', image=image10, height=70, width=290, compound=CENTER, font=('Cambria', 15))
    button10.place(x=800, y=120)
    button13 = Button(swin, text='Back', fg='white', image=image10, height=20, width=75, compound=CENTER, font=('Cambria', 15), command=back1r)

    button13.place(x=1180, y=265)

def search():

    get_l = []

    l=[]

    mycur.execute('select * from s_cust1')

    data2 = mycur.fetchall()

    mycur.execute('select father_name, father_number, mother_name, mother_number, guardian_name, guardian_number, annual_income, 1st_term_fee, 2nd_term_fee from s_cust2')

    data_stu = mycur.fetchall()

    data2 = list(data2)

    for i in range(len(data2)):

        for j in range(len(data_stu)):

            data2[i] = list(data2[i])+list(data_stu[i])

            break

    def find():

        f = 0

        f2 = 0
        l_dict = {}
        for i in tree1.get_children():
            tree1.delete(i)
        get_l = [entry1s.get().strip(), entry2s.get().strip(), combo1s.get().strip(), combo2s.get().strip(), combo3s.get().strip(), entry3s.get().strip(), combo4s.get().strip(), combo5s.get().strip()]
        mycur.execute('select admission_no,name,class,gender,blood_group,mother_tongue from s_cust1')
        data = mycur.fetchall()
        mycur.execute('select 1st_term_fee, 2nd_term_fee from s_cust2')
        data_stu = mycur.fetchall()
        data = list(data)
        for i in range(len(data)):
            for j in range(len(data_stu)):
                data[i] = list(data[i])+list(data_stu[i])
                break
        for i in get_l:
            if i != '':
                l_dict[get_l.index(i)] = i
                index = get_l.index(i)
                get_l.remove(i)
                get_l.insert(index, '')
        for i in data:
            for j in l_dict:
                if str(l_dict[j]).lower() == str(i[j]).lower():
                    f += 1
                if f == len(l_dict):
                    tree1.insert('', END, values=data2[data.index(i)])
                    f2 = 1
                if f != len(l_dict) and f2 != 1:
                    for k in tree1.get_children():
                        tree1.delete(k)
            f = 0
    def reset_s():
        entry1s.delete(0, END)
        entry2s.delete(0, END)
        combo1s.delete(0, END)
        combo2s.delete(0, END)
        combo3s.delete(0, END)
        entry3s.delete(0, END)
```

```
        combo4s.delete(0, END)
        combo5s.delete(0, END)
        for k in tree1.get_children():
            tree1.delete(k)
        for i in data2:
            tree1.insert('', END, values=i)
    button4.destroy()
    button5.destroy()
    button6.destroy()
    button7.destroy()
    label1s = Label(swin, text='Admission Number : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=150, y=50)
    entry1s = ttk.Entry(swin, font=('Cambria', 12, 'bold'))
    entry1s.place(x=330, y=50)
    label2s = Label(swin, text='Name : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=550, y=50)
    entry2s = ttk.Entry(swin, font=('Cambria', 12, 'bold'))
    entry2s.place(x=680, y=50)
    label3s = Label(swin, text='Class : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=895, y=50)
    list1s = ['I', 'II', 'III', 'IV', 'V', 'VI', 'VII', 'VIII', 'IX', 'X', 'XI', 'XII']
    combo1s = ttk.Combobox(swin, value=list1s, font=('Cambria', 12, 'bold'))
    combo1s.place(x=1045, y=50)
    label4s=Label(swin, text='Gender : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=150, y=120)
    list2s = ['Male', 'Female', 'Others']
    combo2s = ttk.Combobox(swin, value=list2s, font=('Cambria', 12, 'bold'))
    combo2s.place(x=330, y=120)
    label5s = Label(swin, text='Blood Group : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=550, y=120)
    list3s = ['O+ve', 'O-ve', 'A+ve', 'A-ve', 'B+ve', 'B-ve', 'AB+ve', 'AB-ve', 'A1+ve', 'Hh']
    combo3s = ttk.Combobox(swin, value=list3s, font=('Cambria', 12, 'bold'))
    combo3s.place(x=680, y=120)
    label6s = Label(swin, text='Mother Tongue : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=895, y=120)
    entry3s = ttk.Entry(swin, font=('Cambria', 12, 'bold'))
    entry3s.place(x=1045, y=120)
    label7s = Label(swin, text='Term I : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=270, y=200)
    list4s = ['Paid', 'Partially Paid', 'Not Paid']

    combo4s = ttk.Combobox(swin, value=list4s, font=('Cambria', 12, 'bold'))

    combo4s.place(x=350, y=200)

    label8s = Label(swin, text='Term II : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=800, y=200)

    combo5s = ttk.Combobox(swin, value=list4s, font=('Cambria', 12, 'bold'))

    combo5s.place(x=893, y=200)

    image9 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg20.png', master=swin)

    image9 = image9.subsample(2, 2)

    button1s = Button(swin, text='Search', font=('Cambria', 12, 'bold'), image=image9, height=25, width=100, bg='black', fg='white', compound=CENTER, command=find)

    button1s.place(x=500, y=250)

    button1s.image = image9

    button2s = Button(swin, text='Reset', font=('Cambria', 12, 'bold'), image=image9, height=25, width=100, bg='black', fg='white', compound=CENTER, command=reset_s)

    button2s.place(x=750, y=250)

    button2s.image = image9
```

```
def update():
    get_l2 = []
    search()
    def item_selected(event):
        global record
        for selected_item in tree1.selection():
            item = tree1.item(selected_item)
            record = item['values']
        swin.withdraw()
        def replace():
            d_entry1.delete(0, END)
            d_entry1.insert(0, record[1])
            entry2.insert(0, record[2])
            combo3.insert(0, record[3])
            d_entry2.delete(0, END)
            d_entry2.insert(0, record[4])
            if record[5] == 'male':
                checkbutton1.select()
            elif record[5] == 'female':
                checkbutton2.select()
```

```
elif record[5] == 'others':
    checkbutton3.select()
combo1.insert(0, record[6])
entry5.insert(0, record[7])
entry6.insert(0, record[8])
entry7.insert(0, record[9])
entry8.insert(0, record[10])
entry9.insert(0, record[11])
entry10.insert(0, record[12])
entry11.insert(0, record[13])
entry12.insert(0, record[14])
if record[15] == 'Paid':
    checkbutton4.select()
elif record[15] == 'Not Paid':
    checkbutton5.select()
elif record[15] == 'Partially Paid':
    checkbutton6.select()
if record[16] == 'Paid':
    checkbutton7.select()
elif record[16] == 'Not Paid':
        checkbutton8.select()
    elif record[16] == 'Partially Paid':
        checkbutton9.select()
def save():
    global gen, fee_1st, fee_2nd
    in_data = list()
    in_values = list()
    in_values1 = list()
    mark_data = list()
    mark_values = list()
    in_data.append(record[0])
    in_values.append('admission_no')
    in_values1.append('admission_no')
    in_data.append(d_entry1.get().strip())
    mark_values.append('admission_no')
    mark_data.append(record[0])
    in_values.append('admission_date')
    if q == 0:
        gen = record[5]
    if h == 0:
        fee_1st = record[15]
    if k == 0:
        fee_2nd = record[16]
    image9 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d1.png', master=add_win)
    image10 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d2.png', master=add_win)
    image11 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d3.png', master=add_win)
    image12 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d4.png', master=add_win)
    image13 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d5.png', master=add_win)
    image14 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d6.png', master=add_win)
    image15 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d7.png', master=add_win)
    image16 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d8.png', master=add_win)
    image17 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d9.png', master=add_win)
    image18 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d10.png', master=add_win)
```

```
        if entry2.get().strip() != '':
            in_data.append(entry2.get().strip())
            in_values.append('name')
            mark_values.append('name')
            mark_data.append(entry2.get().strip())
        else:
            label24 = Label(add_win, text='●', font=('Cambria', 14,
'bold'),  fg='red', height=20, width=10, image=image9,
compound=CENTER)
            label24.image = image9
            label24.place(x=555, y=187)
            add_win.after(2500, label24.destroy)
        if combo3.get().strip() != '':
            in_data.append(combo3.get().strip())
            in_values.append('class')
            mark_values.append('class')
            mark_data.append(combo3.get().strip())
        else:
            label25 = Label(add_win, text='●', font=('Cambria', 14,
'bold'), fg='red', height=20, width=10, image=image9,
compound=CENTER)
```

```
            label25.image = image9
            label25.place(x=570, y=251)
            add_win.after(2500, label25.destroy)
        in_data.append(d_entry2.get().strip())
        in_values.append('dob')
        in_data.append(gen)
        in_values.append('gender')
        if combo1.get().strip() != '':
            in_data.append(combo1.get().strip())
            in_values.append('blood_group')
        else:
            label26 = Label(add_win, text='●', font=('Cambria', 14,
'bold'), fg='red', height=20, width=10, image=image10,
compound=CENTER)
            label26.image = image10
            label26.place(x=570, y=446)
            add_win.after(2500, label26.destroy)
        if entry5.get().strip() != '':
            in_data.append(entry5.get().strip())
            in_values.append('mother_tongue')
        else:
            label27 = Label(add_win, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image11, compound=CENTER)

            label27.image = image11

            label27.place(x=553, y=512)

            add_win.after(2500, label27.destroy)

        if entry6.get() == '' and entry7.get() == '' and entry8.get() == '' and entry9.get() == '' and entry10.get() == '' and entry11.get() == '':

            label28 = Label(add_win, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image13, compound=CENTER)

            label28.image = image13

            label28.place(x=553, y=575)

            add_win.after(2500, label28.destroy)

            label29 = Label(add_win, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image14, compound=CENTER)

            label29.image = image14

            label29.place(x=553, y=640)

            add_win.after(2500, label29.destroy)

            label30 = Label(add_win, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image15, compound=CENTER)
            label30.image = image15
            label30.place(x=1315, y=60)
            add_win.after(2500, label30.destroy)

            label31 = Label(add_win, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image16, compound=CENTER)
            label31.image = image16
            label31.place(x=1315, y=125)
            add_win.after(2500, label31.destroy)

            label32 = Label(add_win, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image17, compound=CENTER)
            label32.image = image17
            label32.place(x=1315, y=190)
            add_win.after(2500, label32.destroy)

            label33 = Label(add_win, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image18, compound=CENTER)
            label33.image = image18
```

```
                label33.place(x=1315, y=255)
                add_win.after(2500, label33.destroy)
            else:
                if entry6.get().strip() != '' and entry7.get().strip() != '':
                    in_data.append(entry6.get().strip())
                    in_values1.append('father_name')
                    if entry7.get().strip().isdigit() and len(entry7.get().strip())
== 10:
                        in_data.append(int(entry7.get().strip()))
                        in_values1.append('father_number')
                    else:
                        label36 = Label(add_win, text='Invalid Phone number !
', font=('Cambria', 12, 'bold'), fg='red', bg='black')
                        label36.place(x=330, y=675)
                        add_win.after(2500, label36.destroy)
                if entry6.get().strip() != '' and entry7.get().strip() == '':
                    in_data.append(entry6.get().strip())
                    in_values1.append('father_name')
                    label29 = Label(add_win, text='●', font=('Cambria', 14,
'bold'), fg='red', height=20, width=10, image=image14,
compound=CENTER)
```

```
                    label29.image = image14

                    label29.place(x=553, y=640)

                    add_win.after(2500, label29.destroy)

                if entry6.get().strip() == '' and entry7.get().strip() != '':

                    if entry7.get().strip().isdigit() and len(entry7.get().strip())
== 10:

                        label28 = Label(add_win, text='●', font=('Cambria', 14,
'bold'), fg='red', height=20, width=10, image=image13,
compound=CENTER)

                        label28.image = image13

                        label28.place(x=553, y=575)

                        add_win.after(2500, label28.destroy)

                    else:

                        label28 = Label(add_win, text='●', font=('Cambria', 14,
'bold'), fg='red', height=20, width=10, image=image13,
compound=CENTER)

                        label28.image = image13

                        label28.place(x=553, y=575)

                        add_win.after(2500, label28.destroy)

                        label36 = Label(add_win, text='Invalid Phone number !
', font=('Cambria', 12, 'bold'), fg='red', bg='black')

                        label36.place(x=330, y=675)
                        add_win.after(2500, label36.destroy)
                if entry8.get().strip() != '' and entry9.get().strip() != '':
                    in_data.append(entry8.get().strip())
                    in_values1.append('mother_name')
                    if entry9.get().strip().isdigit() and len(entry9.get().strip())
== 10:
                        in_data.append(int(entry9.get().strip()))
                        in_values1.append('mother_number')
                    else:
                        label36 = Label(add_win, text='Invalid Phone number !
', font=('Cambria', 12, 'bold'), fg='red', bg='black')
                        label36.place(x=1090, y=160)
                        add_win.after(2500, label36.destroy)
                if entry8.get().strip() != '' and entry9.get().strip() == '':
                    in_data.append(entry8.get().strip())
                    in_values1.append('mother_name')
                    label31 = Label(add_win, text='●', font=('Cambria', 14,
'bold'), fg='red', height=20, width=10, image=image16,
compound=CENTER)
                    label31.image = image16
                    label31.place(x=1315, y=125)
                    add_win.after(2500, label31.destroy)
                if entry8.get().strip() == '' and entry9.get().strip() != '':
                    if entry9.get().strip().isdigit() and len(entry9.get().strip())
== 10:
                        in_data.append(int(entry9.get().strip()))
                        in_values1.append('mother_number')
                        label30 = Label(add_win, text='●', font=('Cambria', 14,
'bold'), fg='red', height=20, width=10, image=image15,
compound=CENTER)
                        label30.image = image15
                        label30.place(x=1315, y=60)
                        add_win.after(2500, label30.destroy)
                    else:
                        label30 = Label(add_win, text='●', font=('Cambria', 14,
'bold'), fg='red', height=20, width=10, image=image15,
compound=CENTER)
                        label30.image = image15
                        label30.place(x=1315, y=60)
                        add_win.after(2500, label30.destroy)
                        label39 = Label(add_win, text='Invalid Phone number !
', font=('Cambria', 12, 'bold'), fg='red', bg='black')
                        label39.place(x=1090, y=160)
                        add_win.after(2500, label39.destroy)
                if entry10.get().strip() != '' and entry11.get().strip() != '':
                    in_data.append(entry10.get().strip())
                    in_values1.append('guardian_name')
                    if entry11.get().strip().isdigit() and
len(entry11.get().strip()) == 10:
                        in_data.append(int(entry11.get().strip()))
                        in_values1.append('guardian_number')
                    else:
                        label36 = Label(add_win, text='Invalid Phone number !
', font=('Cambria', 12, 'bold'), fg='red', bg='black')
                        label36.place(x=1090, y=290)
                        add_win.after(2500, label36.destroy)
                if entry10.get().strip() != '' and entry11.get().strip() == '':
                    in_data.append(entry10.get().strip())
                    in_values1.append('guardian_name')
                    label33 = Label(add_win, text='●', font=('Cambria', 14,
'bold'), fg='red', height=20, width=10, image=image18,
compound=CENTER)
                    label33.image = image18
                    label33.place(x=1315, y=255)
                    add_win.after(2500, label33.destroy)
                if entry10.get().strip() == '' and entry11.get().strip() != '':
                    if entry11.get().strip().isdigit() and len(entry11.get().strip()) == 10:
                        in_data.append(entry11.get().strip())
                        in_values1.append('guardian_number')
                        label32 = Label(add_win, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image17, compound=CENTER)
                        label32.image = image17
                        label32.place(x=1315, y=190)
                        add_win.after(2500, label32.destroy)
                    else:
                        label32 = Label(add_win, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image17, compound=CENTER)
                        label32.image = image17
                        label32.place(x=1315, y=190)
                        add_win.after(2500, label32.destroy)
                        label42 = Label(add_win, text='Invalid Phone number ! ', font=('Cambria', 12, 'bold'), fg='red', bg='black')
                        label42.place(x=1090, y=290)
                        add_win.after(2500, label42.destroy)
            if entry12.get().strip() != '':
                if entry12.get().strip().isalpha() == False :
                    in_data.append(entry12.get())
                    in_values1.append('annual_income')
                else:
                    label34 = Label(add_win, text='Invalid income ! ', font=('Cambria', 12, 'bold'), fg='red', bg='black')
                    label34.place(x=1120, y=355)
                    add_win.after(2500, label34.destroy)
            else:
                label35 = Label(add_win, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image12, compound=CENTER)
                label35.image = image12
                label35.place(x=1315, y=320)
                add_win.after(2500, label35.destroy)
            in_data.append(fee_1st)
            in_values1.append('1st_term_fee')
            in_data.append(fee_2nd)
            in_values1.append('2nd_term_fee')
```

```
          in_data1 = in_data[:8]

          in_data2 = [in_data[0]]+in_data[8:]

          if len(mark_data) == 3 :

             if mark_data[2] in ['XI','XII']:

                mark_data.append('eng,mat,chem,phy,cs')

             else:

                mark_data.append('eng,mat,sci,sco,IIlang')

          if len(in_values) != 0 and len(in_data) != 0 and len(in_values1)
!= 0 and len(in_data2) != 0 and len(mark_values) != 0 and
len(mark_data) != 0:

             if len(in_values) == len(in_data1) and len(in_values1) ==
len(in_data2) and len(mark_values) == len(mark_data) :

                code0 = 'delete from s_cust1 where admission_no =
'+str(in_data[0])

                code = 'delete from s_cust2 where admission_no =
'+str(in_data[0])

                code1 = 'delete from class_det where admission_no =
'+str(in_data[0])

                code2 = 'insert into s_cust1
'+str(tuple(in_values)).replace("'",'')+' values '+str(tuple(in_data1))

                code3 = 'insert into s_cust2
'+str(tuple(in_values1)).replace("'",'')+' values
'+str(tuple(in_data2))
```

```
            code4 = 'insert into class_det
'+str(tuple(mark_values)).replace("'",'')+' values
'+str(tuple(mark_data))

            mycur.execute(code0)

            mycur.execute(code)

            mycur.execute(code1)

            mycur.execute(code2)

            mycur.execute(code3)

            mycur.execute(code4)

            mycon.commit()

            add_win.withdraw()

            if messagebox.showinfo('Info','Student details
successfully saved .'):

                stu_cust()

    def reset():

        d_entry1.delete(0, END)

        d_entry1.insert(0, date1)

        entry2.delete(0, END)

        entry2.insert(0, '')

        combo3.delete(0, END)

        combo3.insert(0, '')
```

```
d_entry2.delete(0, END)
d_entry2.insert(0, date1)
checkbutton1.deselect()
checkbutton2.deselect()
checkbutton3.deselect()
combo1.delete(0, END)
combo1.insert(0, '')
entry5.delete(0, END)
entry5.insert(0, '')
entry6.delete(0, END)
entry6.insert(0, '')
entry7.delete(0, END)
entry7.insert(0, '')
entry8.delete(0, END)
entry8.insert(0, '')
entry9.delete(0, END)
entry9.insert(0, '')
entry10.delete(0, END)
entry10.insert(0, '')
entry11.delete(0, END)
```

```
    entry11.insert(0, '')
    entry12.delete(0, END)
    entry12.insert(0, '')
    if x == 1:
        if fee_1st == 'Partially Paid':
            back1()
    if y == 1:
        if fee_2nd == 'Partially Paid':
            back2()
    checkbutton4.deselect()
    checkbutton5.deselect()
    checkbutton6.deselect()
    checkbutton7.deselect()
    checkbutton8.deselect()
    checkbutton9.deselect()
def cancel():
    add_win.withdraw()
    if messagebox.askyesno('', 'Are you sure to Exit ?'):
        stu_cust()
add_win = Tk()
```

```
    add_win.title('Update students data')

    add_win.state('zoomed')

    icon16 = PhotoImage(file=r'C:\Users\User\Desktop\New
folder\office5.png', master=add_win)

    add_win.iconphoto(False, icon16)

    image7 = PhotoImage(file=r'C:\Users\User\Desktop\New
folder\bg18.png', master=add_win)

    label23 = Label(add_win, image=image7).place(x=-2, y=-2)

    a = str(int(data[-1][0]) + 1)

    label3 = Label(add_win, text='UPDATE STUDENT DATA',
font=('Cambria', 20), bg='black', fg='gray').pack()

    label4 = Label(add_win, text='Admission Number : ',
font=('Cambria', 15, 'bold'), bg='black', fg='white').place(

        x=50, y=60)

    label5 = Label(add_win, text=record[0], font=('Cambria', 15,
'bold'), bg='black', fg='cyan').place(x=300, y=60)

    label6 = Label(add_win, text='Admission Date : ',
font=('Cambria', 15, 'bold'), bg='black', fg='white').place(

        x=50, y=120)

    d_entry1 = DateEntry(add_win, font=('Cambria', 15, 'bold'),
width=20, selectmode='day', maxdate=date.today(),
                    background='black', foreground='white',
date_pattern='dd/mm/y')

    d_entry1.place(x=300, y=121)

    date1 = d_entry1.get()

    dob = date1.replace(date1[6:], str(int(date1[6:]) - 4))

    dob = date(int(dob[6:]), int(dob[3:5]), int(dob[:2]))

    d_entry1.focus()

    label7 = Label(add_win, text='Name of the Student : ',
font=('Cambria', 15, 'bold'), bg='black',

                fg='white').place(x=50, y=185)

    entry2 = ttk.Entry(add_win, font=('Cambria', 15, 'bold'))

    entry2.place(x=300, y=186)

    label8 = Label(add_win, text='Class : ', font=('Cambria', 15,
'bold'), bg='black', fg='white').place(x=50, y=250)

    list3 = ['I', 'II', 'III', 'IV', 'V', 'VI', 'VII', 'VIII', 'IX', 'X', 'XI', 'XII']

    combo3 = ttk.Combobox(add_win, value=list3, font=('Cambria',
15, 'bold'))

    combo3.place(x=300, y=251)

    label9 = Label(add_win, text='Date of Birth : ', font=('Cambria',
15, 'bold'), bg='black', fg='white').place(x=50, y=315)
```

```
    d_entry2 = DateEntry(add_win, font=('Cambria', 15, 'bold'), width=20, selectmode='day', maxdate=dob, background='black', foreground='white', date_pattern='dd/mm/y')
    d_entry2.place(x=300, y=316)
    label10 = Label(add_win, text='Gender : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=50, y=380)
    def male():
        global gen, q
        q = 1
        gen = 'male'
    def female():
        global gen, q
        q = 2
        gen = 'female'
    def others():
        global gen, q
        q = 3
        gen = 'others'
    checkbutton1 = Checkbutton(add_win, text='Male', bg='lightblue', activebackground='lightblue', height=1,
                        variable=1, onvalue=1, offvalue=0,
font=('Cambria', 12, 'bold'), command=male)

    checkbutton1.place(x=300, y=380)

    checkbutton2 = Checkbutton(add_win, text='Female',
bg='lightblue', activebackground='lightblue', height=1,

                        variable=1, onvalue=2, offvalue=0,
font=('Cambria', 12, 'bold'), command=female)

    checkbutton2.place(x=400, y=380)

    checkbutton3 = Checkbutton(add_win, text='Others',
bg='lightblue', activebackground='lightblue', height=1,  variable=1,
onvalue=3, offvalue=0, font=('Cambria', 12, 'bold'),
command=others)

    checkbutton3.place(x=525, y=380)

    label11 = Label(add_win, text='Blood Group : ', font=('Cambria',
15, 'bold'), bg='black', fg='white').place(

      x=50, y=445)

    list1 = ['O+ve', 'O-ve', 'A+ve', 'A-ve', 'B+ve', 'B-ve', 'AB+ve', 'AB-
ve', 'A1+ve', 'Hh']

    combo1 = ttk.Combobox(add_win, value=list1, font=('Cambria',
15, 'bold'))

    combo1.place(x=300, y=446)

    label12 = Label(add_win, text='Mother Tongue : ',
font=('Cambria', 15, 'bold'), bg='black', fg='white').place(
      x=50, y=510)
    entry5 = ttk.Entry(add_win, font=('Cambria', 15, 'bold'))
    entry5.place(x=300, y=511)
    label13 = Label(add_win, text="Father's Name : ",
font=('Cambria', 15, 'bold'), bg='black', fg='white').place(
      x=50, y=575)
    entry6 = ttk.Entry(add_win, font=('Cambria', 15, 'bold'))
    entry6.place(x=300, y=576)
    label14 = Label(add_win, text="Father's Phone Number : ",
font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=50,
y=640)
    entry7 = ttk.Entry(add_win, font=('Cambria', 15, 'bold'))
    entry7.place(x=300, y=641)
    label15 = Label(add_win, text="Mother's Name : ",
font=('Cambria', 15, 'bold'), bg='black', fg='white').place(
      x=800, y=60)
    entry8 = ttk.Entry(add_win, font=('Cambria', 15, 'bold'))
    entry8.place(x=1060, y=61)
    label16 = Label(add_win, text="Mother's Phone Number : ",
font=('Cambria', 15, 'bold'), bg='black',
                fg='white').place(x=800, y=125)
```

```
    entry9 = ttk.Entry(add_win, font=('Cambria', 15, 'bold'))
    entry9.place(x=1060, y=126)
    label17 = Label(add_win, text="Guardian Name : ",
font=('Cambria', 15, 'bold'), bg='black', fg='white').place(
        x=800, y=190)
    entry10 = ttk.Entry(add_win, font=('Cambria', 15, 'bold'))
    entry10.place(x=1060, y=191)
    label18 = Label(add_win, text="Guardian Phone Number : ",
font=('Cambria', 15, 'bold'), bg='black',
                    fg='white').place(x=800, y=255)
    entry11 = ttk.Entry(add_win, font=('Cambria', 15, 'bold'))
    entry11.place(x=1060, y=256)
    label19 = Label(add_win, text="Annual Income : ",
font=('Cambria', 15, 'bold'), bg='black', fg='white').place(
        x=800, y=320)
    entry12 = ttk.Entry(add_win, font=('Cambria', 15, 'bold'))
    entry12.place(x=1060, y=321)
    def paid1():
        global fee_1st, h
        h = 1
        fee_1st = 'Paid'
```

```
def not_paid1():
    global fee_1st, h
    h = 2
    fee_1st = 'Not Paid'
def partially_paid1():
    global fee_1st, back1, x, h
    def back1():
        button21.destroy()
        label21.destroy()
        entry13.destroy()
        checkbutton6.deselect()
        checkbutton4['state'] = ACTIVE
        checkbutton5['state'] = ACTIVE
        checkbutton6['state'] = ACTIVE
    x = 1
    h = 3
    fee_1st = 'Partially Paid'
    checkbutton4['state'] = DISABLED
    checkbutton5['state'] = DISABLED
    checkbutton6['state'] = DISABLED
```

```
        image8 = PhotoImage(file=r'C:\Users\User\Desktop\New
folder\back1.png', master=add_win)

        button21 = Button(add_win, width=20, height=20,
image=image8, bd=0, command=back1, bg='black',

                      activebackground='lightblue')

        button21.place(x=750, y=463)

        button21.image = image8

        label21 = Label(add_win, text='Balance Amount : ',
font=('Cambria', 15, 'bold'), bg='black', fg='white')

        label21.place(x=800, y=460)

        entry13 = ttk.Entry(add_win, font=('Cambria', 15, 'bold'))

        entry13.place(x=1060, y=461)

        entry13.focus()

      label20 = Label(add_win, text='First Term Fees : ',
font=('Cambria', 15, 'bold'), bg='black', fg='white').place(

        x=800, y=385)

      checkbutton4 = Checkbutton(add_win, text='Paid',
bg='lightblue', activebackground='lightblue', height=1,

                         variable=2, onvalue=1, offvalue=0,
font=('Cambria', 12, 'bold'), command=paid1)

      checkbutton4.place(x=1060, y=385)
```

```
    checkbutton5 = Checkbutton(add_win, text='Not Paid',
bg='lightblue', activebackground='lightblue', height=1,
                             variable=2, onvalue=2, offvalue=0,
font=('Cambria', 12, 'bold'), command=not_paid1)
    checkbutton5.place(x=1200, y=385)
    checkbutton6 = Checkbutton(add_win, text='Partially Paid',
bg='lightblue', activebackground='lightblue',
                             height=1, variable=2, onvalue=3, offvalue=0,
font=('Cambria', 12, 'bold'),
                             command=partially_paid1)
    checkbutton6.place(x=1100, y=422)
    def paid2():
        global fee_2nd, k
        k = 1
        fee_2nd = 'Paid'
    def not_paid2():
        global fee_2nd, k
        k = 2
        fee_2nd = 'Not Paid'
    def partially_paid2():
        global fee_2nd, back2, y, k
```

```
def back2():
    button20.destroy()
    label23.destroy()
    entry14.destroy()
    checkbutton9.deselect()
    checkbutton7['state'] = ACTIVE
    checkbutton8['state'] = ACTIVE
    checkbutton9['state'] = ACTIVE
y = 1
k = 3
fee_2nd = 'Partially Paid'
checkbutton7['state'] = DISABLED
checkbutton8['state'] = DISABLED
checkbutton9['state'] = DISABLED
image7 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\back1.png', master=add_win)
button20 = Button(add_win, width=20, height=20, image=image7, bd=0, command=back2, bg='black',
                  activebackground='lightblue')
button20.place(x=750, y=603)
button20.image = image7
```

```
            label23 = Label(add_win, text='Balance Amount : ',
font=('Cambria', 15, 'bold'), bg='black', fg='white')

            label23.place(x=800, y=600)

            entry14 = ttk.Entry(add_win, font=('Cambria', 15, 'bold'))

            entry14.place(x=1060, y=601)

            entry14.focus()

        label22 = Label(add_win, text='Second Term Fees : ',
font=('Cambria', 15, 'bold'), bg='black',

                        fg='white').place(x=800, y=525)

        checkbutton7 = Checkbutton(add_win, text='Paid',
bg='lightblue', activebackground='lightblue', height=1,

                                    variable=3, onvalue=1, offvalue=0,
font=('Cambria', 12, 'bold'), command=paid2)

        checkbutton7.place(x=1060, y=525)

        checkbutton8 = Checkbutton(add_win, text='Not Paid',
bg='lightblue', activebackground='lightblue', height=1,

                                    variable=3, onvalue=2, offvalue=0,
font=('Cambria', 12, 'bold'), command=not_paid2)

        checkbutton8.place(x=1200, y=525)

        checkbutton9 = Checkbutton(add_win, text='Partially Paid',
bg='lightblue', activebackground='lightblue',
                                    height=1, variable=3, onvalue=3, offvalue=0, font=('Cambria', 12, 'bold'),
                                    command=partially_paid2)
        checkbutton9.place(x=1100, y=562)
        image9 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg20.png', master=add_win)
        image9 = image9.subsample(2, 2)
        button13 = Button(add_win, text='Save', font=('Cambria', 15, 'bold'), image=image9, fg='white', height=30,
                        width=120, command=save, compound=CENTER)
        button13.place(x=800, y=660)
        button14 = Button(add_win, text='Reset', font=('Cambria', 15, 'bold'), image=image9, fg='white', height=30,
                        width=120, command=reset, compound=CENTER)
        button14.place(x=1000, y=660)
        button15 = Button(add_win, text='Cancel', font=('Cambria', 15, 'bold'), image=image9, fg='white', height=30,
                        width=120, command=cancel, compound=CENTER)
        button15.place(x=1200, y=660)
        replace()
        add_win.mainloop()
    tree1.bind('<Double-1>', item_selected)
def teach_cust():
    global twin, tree2, button_tc2, button_tc3, button_tc4, button_tc5,
data_teach
    owin.withdraw()
    mycur.execute('select * from t_cust')
    data_teach = mycur.fetchall()
    mycur.execute('select qualification, subject, class_teaching1,
class_teaching2, class_teacher, designation, salary_status from
t_cust1')
    data_t = mycur.fetchall()
    data_teach = list(data_teach)
    for i in range(len(data_teach)):
        for j in range(len(data_t)):
            data_teach[i] = list(data_teach[i])+list(data_t[i])
            break
    data_teach2 = []
    twin = Tk()
    twin.state('zoomed')
    twin.title('Teacher Customization')
    icon17 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\office5.png', master=twin)

    twin.iconphoto(False, icon17)

    image_tc1 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg3.png', master=twin)

    image_tc2 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\home4.png', master=twin)

    image_tc2 = image_tc2.subsample(3, 3)

    label_tc1 = Label(twin, image=image_tc1).place(x=-3, y=-3)

    image11 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg20.png', master=twin)

    image11 = image11.subsample(1, 1)

    button_tc1 = Button(twin, image=image_tc2, height=30, width=30, bd=0, bg='black', activebackground='black', command=home_teach)

    button_tc1.place(x=10, y=10)

    frame2 = Frame(twin, height=300, width=1200)

    frame2.pack()

    image_tc3 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg4.png', master=frame2)

    label_tc2 = Label(frame2, image=image_tc3).place(x=-5, y=-5)

    label_tc0 = Label(frame2, text='Teacher Customization', font=('Cambria', 20), bg='black', fg='gray').place(x=455, y=0)
    button_tc2 = Button(frame2, text='ADD', image=image11, height=32, width=160, compound=CENTER, fg='white', font=('Cambria', 15, 'bold'), command=add_teach)
    button_tc2.place(x=100, y=90)
    button_tc3 = Button(frame2, text='UPDATE', image=image11, height=32, width=160, compound=CENTER, fg='white', font=('Cambria', 15, 'bold'), command=update_teach)
    button_tc3.place(x=900, y=90)
    button_tc4 = Button(frame2, text='REMOVE', image=image11, height=32, width=160, compound=CENTER, fg='white', font=('Cambria', 15, 'bold'), command=remove_teach)
    button_tc4.place(x=280, y=200)
    button_tc5 = Button(frame2, text='SEARCH', image=image11, height=32, width=160, compound=CENTER, fg='white', font=('Cambria', 15, 'bold'), command=search_teach)
    button_tc5.place(x=720, y=200)
    style = ttk.Style(twin)
    style.theme_use('vista')
    style.configure('Treeview', rowheight=28, background='lightblue', foreground='black', font=('Cambria', 10))
    style.configure('Treeview.Heading', font=('Cambria', 10, 'bold'))
    sb2_t = ttk.Scrollbar(twin, orient="vertical")
    sb2_t.pack(side='right', fill='y')
    col = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)
    tree2 = ttk.Treeview(twin, column=col, show='headings',
height=13)
    tree2.pack()
    sb1_t = ttk.Scrollbar(twin, orient="horizontal",
command=tree2.xview)
    sb1_t.pack(side='bottom', fill='x')
    tree2.configure(xscrollcommand=sb1_t.set)
    tree2.configure(yscrollcommand=sb2_t.set)
    sb2_t.configure(command=tree2.yview)
    tree2.column(1, anchor=CENTER, stretch=NO, width=100)
    tree2.column(2, anchor=CENTER, stretch=NO, width=120)
    tree2.column(3, anchor=CENTER, stretch=NO, width=150)
    tree2.column(4, anchor=CENTER, stretch=NO, width=100)
    tree2.column(5, anchor=CENTER, stretch=NO, width=120)
    tree2.column(6, anchor=CENTER, stretch=NO, width=120)
    tree2.column(7, anchor=CENTER, stretch=NO, width=100)
    tree2.column(8, anchor=CENTER, stretch=NO, width=120)
    tree2.column(9, anchor=CENTER, stretch=NO, width=150)
    tree2.column(10, anchor=CENTER, stretch=NO, width=150)
    tree2.column(11, anchor=CENTER, stretch=NO, width=150)
```

```
tree2.column(12, anchor=CENTER, stretch=NO, width=150)
tree2.column(13, anchor=CENTER, stretch=NO, width=150)
tree2.column(14, anchor=CENTER, stretch=NO, width=150)
tree2.column(15, anchor=CENTER, stretch=NO, width=150)
tree2.heading(1, text='Teacher Id')
tree2.heading(2, text='Joining date')
tree2.heading(3, text='Name')
tree2.heading(4, text='Date of the Birth')
tree2.heading(5, text='Gender')
tree2.heading(6, text='Blood group')
tree2.heading(7, text='Mother Tongue')
tree2.heading(8, text='Phone.no')
tree2.heading(9, text='Qualification')
tree2.heading(10, text='subject')
tree2.heading(11, text='Classes Teaching')
tree2.heading(12, text='Classes Teaching')
tree2.heading(13, text='Class Teacher Of')
tree2.heading(14, text='Designation')
tree2.heading(15, text='Salary Status')
for j in range(len(data_teach)):
        for i in data_teach[j]:
            data_teach2.append(i)
        tree2.insert('', END, values=data_teach2)
        data_teach2.clear()
    def close():
        twin.withdraw()
        office_func()
    twin.protocol('WM_DELETE_WINDOW',close)
    twin.mainloop()
def home_teach():
    twin.withdraw()
    office_func()
def add_teach():
    def save():
        in_data_t = list()
        in_values_t1 = list()
        in_values_t2 = list()
        login_data = list()
        login_values = list()
        in_data_t.append(a)
        in_values_t1.append('teacher_id')
        in_values_t2.append('teacher_id')
        in_data_t.append(d_entry1.get())
        in_values_t1.append('joining_date')
        image9 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d1.png', master=addt_wn)
        image10 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d2.png', master=addt_wn)
        image11 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d3.png', master=addt_wn)
        image12 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d4.png', master=addt_wn)
        image13 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d5.png', master=addt_wn)
        image14 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d6.png', master=addt_wn)
        image15 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d7.png', master=addt_wn)
        image16 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d8.png', master=addt_wn)
        image17 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d9.png', master=addt_wn)
        image18 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d10.png', master=addt_wn)

        image19 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d11.png', master=addt_wn)

        image20 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d12.png', master=addt_wn)

        if entry2.get().strip() != '':

            in_data_t.append(entry2.get().strip().title())

            in_values_t1.append('name')

            login_values.append('un')

            login_data.append(entry2.get().strip().title())

            login_values.append('up')

            login_data.append(entry2.get().strip().title().replace(' ','')+a)

        else:

            label24 = Label(addt_wn, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image9, compound=CENTER)

            label24.image = image9

            label24.place(x=555, y=217)

            addt_wn.after(2500, label24.destroy)

        in_data_t.append(d_entry2.get())

        in_values_t1.append('dob')
        in_data_t.append(gen)
        in_values_t1.append('gender')
        if combo1.get().strip() != '':
            in_data_t.append(combo1.get())
            in_values_t1.append('blood_group')
        else:
            label26 = Label(addt_wn, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image10, compound=CENTER)
            label26.image = image10
            label26.place(x=570, y=455)
            addt_wn.after(2500, label26.destroy)
        if entry5.get().strip() != '':
            in_data_t.append(entry5.get())
            in_values_t1.append('mother_tongue')
        else:
            label27 = Label(addt_wn, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image11, compound=CENTER)
            label27.image = image11
            label27.place(x=553, y=537)
```

```
            addt_wn.after(2500, label27.destroy)
        if entry11.get() == '':
            label33 = Label(addt_wn, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image18, compound=CENTER)
            label33.image = image18
            label33.place(x=1305, y=57)
            addt_wn.after(2500, label33.destroy)
        else:
            if entry11.get().strip() != '':
                if entry11.get().strip().isdigit() and len(entry11.get().strip()) == 10:
                    in_data_t.append(entry11.get().strip())
                    in_values_t1.append('phone_no')
                else:
                    label36 = Label(addt_wn, text='Invalid Phone number ! ', font=('Cambria', 12, 'bold'), fg='red', bg='black')
                    label36.place(x=1080, y=95)
                    addt_wn.after(2500, label36.destroy)
        if entry12.get().strip() != '':
            in_data_t.append(entry12.get())
            in_values_t2.append('qualification')
        else:
            label35 = Label(addt_wn, text='●', font=('Cambria', 14, 'bold'),
    fg='red', height=20, width=10, image=image12,
    compound=CENTER)
            label35.image = image12
            label35.place(x=1305, y=138)
            addt_wn.after(2500, label35.destroy)
        if entry13.get().strip() != '':
            in_data_t.append(entry13.get().strip())
            in_values_t2.append('subject')
        if combo2.get().strip() != '':
            in_data_t.append(combo2.get().strip())
            in_values_t2.append('class_teaching1')
        if combo3.get().strip() != '':
            in_data_t.append(combo3.get().strip())
            in_values_t2.append('class_teaching2')
        if combo4.get().strip() != '':
            in_data_t.append(combo4.get().strip())
            in_values_t2.append('class_teacher')
            login_values.append('class')
            login_data.append(combo4.get().strip())
        if entry14.get().strip() != '':
            in_data_t.append(entry14.get().strip())
            in_values_t2.append('designation')
            login_values.append('ut')
            login_data.append(entry14.get().strip())
        else:
            label38 = Label(addt_wn, text='●', font=('Cambria', 14, 'bold'),
    fg='red', height=20, width=10, image=image19,
    compound=CENTER)
            label38.image = image19
            label38.place(x=1305, y=457)
            addt_wn.after(2500, label38.destroy)
        if entry15.get().strip() != '':
            in_data_t.append(entry15.get().strip())
            in_values_t2.append('salary_status')
        else:
            label37 = Label(addt_wn, text='●', font=('Cambria', 14, 'bold'),
    fg='red', height=20, width=10, image=image20,
    compound=CENTER)
            label37.image = image20
```

```
          label37.place(x=1305, y=540)

          addt_wn.after(2500, label37.destroy)

        in_values_t2.append('up')

        in_data_t1 = in_data_t[:8]

        in_data_t2 = [in_data_t[0]]+in_data_t[8:]

        in_data_t2.append(entry2.get().strip().title().replace(' ','')+a)

        if len(in_values_t1) != 0 and len(in_data_t1) != 0 and
len(in_values_t2) != 0 and len(in_data_t2) != 0 and len(login_values)
!= 0 and len(login_data) != 0:

          if len(in_values_t1) == len(in_data_t1) and len(in_values_t2)
== len(in_data_t2) and len(login_values) == len(login_data) :

            code1 = 'insert into t_cust
'+str(tuple(in_values_t1)).replace("'",'')+' values
'+str(tuple(in_data_t1))

            code2 = 'insert into t_cust1
'+str(tuple(in_values_t2)).replace("'",'')+' values
'+str(tuple(in_data_t2))

            code3 = 'insert into login
'+str(tuple(login_values)).replace("'",'')+' values
'+str(tuple(login_data))

            mycur.execute(code1)

            mycur.execute(code2)

            mycur.execute(code3)
```

```
                mycur.execute('commit')

                addt_wn.withdraw()

                if messagebox.showinfo('Info','Teacher details successfully
saved .'):

                    teach_cust()

    def reset():

        d_entry1.delete(0, END)

        d_entry1.insert(0, date1)

        entry2.delete(0, END)

        entry2.insert(0, '')

        d_entry2.delete(0, END)

        dob2 = str(date.today())[8:] + '/' + str(date.today())[5:7] + '/' +
str(int(str(date.today())[:4]) - 4)

        d_entry2.insert(0, dob2)

        checkbutton1.deselect()

        checkbutton2.deselect()

        checkbutton3.deselect()

        combo1.delete(0, END)

        combo1.insert(0, '')

        combo2.delete(0, END)

        combo2.insert(0, '')
        combo3.delete(0, END)
        combo3.insert(0, '')
        combo4.delete(0, END)
        combo4.insert(0, '')
        entry5.delete(0, END)
        entry5.insert(0, '')
        entry11.delete(0, END)
        entry11.insert(0, '')
        entry12.delete(0, END)
        entry12.insert(0, '')
        entry13.delete(0, END)
        entry13.insert(0, '')
        entry14.delete(0, END)
        entry14.insert(0, '')
        entry15.delete(0, END)
        entry15.insert(0, '')
    def cancel():
        addt_wn.withdraw()
        if messagebox.askyesno('', 'Are you sure to Exit ?'):
            teach_cust()
    global image9

    twin.withdraw()

    addt_wn = Tk()

    addt_wn.title('Append students data')

    addt_wn.state('zoomed')

    icon11 = PhotoImage(file=r'C:\Users\User\Desktop\New
folder\office5.png', master=addt_wn)

    addt_wn.iconphoto(False, icon11)

    image7 = PhotoImage(file=r'C:\Users\User\Desktop\New
folder\bg18.png', master=addt_wn)

    label23 = Label(addt_wn, image=image7).place(x=-2, y=-2)

    a = '#'+str(int(data_teach[-1][0][1:])+1)

    label3 = Label(addt_wn, text='ADD DATA', font=('Cambria', 20),
bg='black', fg='gray').pack()

    label4 = Label(addt_wn, text='Teacher Id : ', font=('Cambria', 15,
'bold'), bg='black', fg='white').place(x=50, y=57)

    label5 = Label(addt_wn, text='  '+a+'  ', font=('Cambria', 15, 'bold'),
bg='black', fg='cyan').place(x=300, y=57)

    label6 = Label(addt_wn, text='Joining Date : ', font=('Cambria', 15,
'bold'), bg='black', fg='white').place(x=50, y=137)
```

```
    d_entry1 = DateEntry(addt_wn, font=('Cambria', 15, 'bold'), width=20, selectmode='day', maxdate=date.today(), background='black', foreground='white', date_pattern='dd/mm/y')

    d_entry1.place(x=300, y=137)

    date1 = d_entry1.get()

    dob = date1.replace(date1[6:], str(int(date1[6:])-20))

    dob = date(int(dob[6:]), int(dob[3:5]), int(dob[:2]))

    d_entry1.focus()

    label7 = Label(addt_wn, text='Name of the Teacher : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=50, y=217)

    entry2 = ttk.Entry(addt_wn, font=('Cambria', 15, 'bold'))

    entry2.place(x=300, y=217)

    label9 = Label(addt_wn, text='Date of Birth : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=50, y=297)

    d_entry2 = DateEntry(addt_wn, font=('Cambria', 15, 'bold'), width=20, selectmode='day', maxdate=dob, background='black', foreground='white', date_pattern='dd/mm/y')

    d_entry2.place(x=300, y=297)

    label10 = Label(addt_wn, text='Gender : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=50, y=377)

    def male():
        global gen
        gen = 'male'
    def female():
        global gen
        gen = 'female'
    def others():
        global gen
        gen = 'others'
    checkbutton1 = Checkbutton(addt_wn, text='Male', bg='lightblue', activebackground='lightblue', height=1, variable=1, onvalue=1, offvalue=0, font=('Cambria', 12, 'bold'), command=male)
    checkbutton1.place(x=300, y=377)
    checkbutton2 = Checkbutton(addt_wn, text='Female', bg='lightblue', activebackground='lightblue', height=1, variable=1, onvalue=2, offvalue=0, font=('Cambria', 12, 'bold'), command=female)
    checkbutton2.place(x=400, y=377)
    checkbutton3 = Checkbutton(addt_wn, text='Others', bg='lightblue', activebackground='lightblue', height=1, variable=1, onvalue=3, offvalue=0, font=('Cambria', 12, 'bold'), command=others)
    checkbutton3.place(x=525, y=377)
    label11 = Label(addt_wn, text='Blood Group : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=50, y=457)

    list1 = ['O+ve', 'O-ve', 'A+ve', 'A-ve', 'B+ve', 'B-ve', 'AB+ve', 'AB-ve', 'A1+ve', 'Hh']

    combo1 = ttk.Combobox(addt_wn, value=list1, font=('Cambria', 15, 'bold'))

    combo1.place(x=300, y=457)

    label12 = Label(addt_wn, text='Mother Tongue : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=50, y=537)

    entry5 = ttk.Entry(addt_wn, font=('Cambria', 15, 'bold'))

    entry5.place(x=300, y=537)

    label18 = Label(addt_wn, text="Phone Number : ", font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=800, y=57)

    entry11 = ttk.Entry(addt_wn, font=('Cambria', 15, 'bold'))

    entry11.place(x=1050, y=57)

    label19 = Label(addt_wn, text="Qualification : ", font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=800, y=137)

    entry12 = ttk.Entry(addt_wn, font=('Cambria', 15, 'bold'))

    entry12.place(x=1050, y=137)

    label20 = Label(addt_wn, text='Subject : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=800, y=217)
    entry13 = ttk.Entry(addt_wn, font=('Cambria', 15, 'bold'))
    entry13.place(x=1050, y=217)
    label20 = Label(addt_wn, text='Class Teaching : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=800, y=297)
    list3 = ['None','I', 'II', 'III', 'IV', 'V', 'VI', 'VII', 'VIII', 'IX', 'X', 'XI', 'XII']
    combo2 = ttk.Combobox(addt_wn, value=list3, font=('Cambria', 15, 'bold'), width=5, height=10)
    combo2.place(x=1050, y=297)
    combo3 = ttk.Combobox(addt_wn, value=list3, font=('Cambria', 15, 'bold'), width=5, height=10)
    combo3.place(x=1200, y=297)
    label20 = Label(addt_wn, text='Class Teacher : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=800, y=377)
    list4 = ['None','I', 'II', 'III', 'IV', 'V', 'VI', 'VII', 'VIII', 'IX', 'X', 'XI', 'XII']
    combo4 = ttk.Combobox(addt_wn, value=list4, font=('Cambria', 15, 'bold'))
    combo4.place(x=1050, y=377)
    label20 = Label(addt_wn, text='Designation : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=800, y=457)
    entry14 = ttk.Entry(addt_wn, font=('Cambria', 15, 'bold'))
    entry14.place(x=1050, y=457)
    label20 = Label(addt_wn, text='Salary Status : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=800, y=537)

    entry15 = ttk.Entry(addt_wn, font=('Cambria', 15, 'bold'))

    entry15.place(x=1050, y=537)

    image9 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg20.png', master=addt_wn)

    image9 = image9.subsample(2, 2)

    button13 = Button(addt_wn, text='Save', font=('Cambria', 15, 'bold'), image=image9, fg='white', height=30, width=120, command=save, compound=CENTER)

    button13.place(x=800, y=660)

    button14 = Button(addt_wn, text='Reset', font=('Cambria', 15, 'bold'), image=image9, fg='white', height=30, width=120, command=reset, compound=CENTER)

    button14.place(x=1000, y=660)

    button15 = Button(addt_wn, text='Cancel', font=('Cambria', 15, 'bold'), image=image9, fg='white', height=30, width=120, command=cancel, compound=CENTER)

    button15.place(x=1200, y=660)

    addt_wn.mainloop()

def remove_teach():

    for i in [button_tc2, button_tc3, button_tc4, button_tc5]:
        i.destroy()
    twin.title('Remove students data')
    image10 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg20.png', master=twin)
    image10 = image10.subsample(1, 1)
    lable9 = Label(twin, text='Teacher ID : ', font=('Cambria', 15), bg='black', fg='gray').place(x=150, y=125)
    lable10 = Label(twin, text='Name of the Teacher : ', font=('Cambria', 15), bg='black', fg='gray').place(x=760, y=125)
    entry1 = ttk.Entry(twin, font=('Cambria', 15, 'bold'))
    entry1.place(x=300, y=125)
    entry2 = ttk.Entry(twin, font=('Cambria', 15, 'bold'))
    entry2.place(x=970, y=125)
    def show_single_t():
        global list_rt
        list_rt =[]
        if entry1.get() != '' and entry2.get() == '':
            for i in tree2.get_children():
                tree2.delete(i)
            mycur.execute('select * from t_cust')
            data = mycur.fetchall()
```

```
mycur.execute('select qualification, subject, class_teaching1, class_teaching2, class_teacher, designation, salary_status from t_cust1')
data_t = mycur.fetchall()
data = list(data)
for i in range(len(data)):
    for j in range(len(data_t)):
        data[i] = list(data[i])+list(data_t[i])
        break
flag1 = 0
for i in data:
    if entry1.get() == i[0]:
        tree2.insert('', END, values=i)
        list_rt.append(i)
        def delete_single_t():
            twin.withdraw()
            m1 = messagebox.askyesno('', 'Are you sure to Delete ?')
            if m1 == True:
                for i in list_rt:
```

```
                code5 = 'delete from t_cust where teacher_id = "'+str(i[0])+'"'
                code6 = 'delete from t_cust1 where teacher_id = "'+str(i[0])+'"'
                code7 = 'delete from login where un = "'+i[2]+'"'
                mycur.execute(code5)
                mycur.execute(code6)
                mycur.execute(code7)
                mycon.commit()
                if messagebox.showinfo('Info','Teacher details successfully removed .'):
                    teach_cust()
            else:
                teach_cust()
        entry1['state'] = DISABLED
        entry2['state'] = DISABLED
        button12 = Button(twin, text='Delete', font=('Cambria', 15, 'bold'), fg='white', image=image10, height=25, width=100, compound=CENTER, command=delete_single_t)
        button12.place(x=623, y=225)
    else:
```

```
                    flag1+=1
            if flag1 == len(data):
                label12 = Label(twin, text='Not Found', font=('Cambria', 12,
'bold'), fg='red', bg='black')
                label12.place(x=637, y=185)
                twin.after(2500, label12.destroy)
        elif entry1.get() == '' and entry2.get() != '':
            for i in tree2.get_children():
                tree2.delete(i)
            mycur.execute('select * from t_cust')
            data = mycur.fetchall()
            mycur.execute('select qualification, subject, class_teaching1,
class_teaching2, class_teacher, designation, salary_status from
t_cust1')
            data_t = mycur.fetchall()
            data = list(data)
            for i in range(len(data)):
                for j in range(len(data_t)):
                    data[i] = list(data[i])+list(data_t[i])
                    break
            flag2 = 0
```

```
      for i in data:
        if entry2.get() == i[2]:
          tree2.insert('', END, values=i)
          list_rt.append(i)
          def delete_single_t():
            twin.withdraw()
            m1 = messagebox.askyesno('', 'Are you sure to Delete
?')
            if m1 == True:
              for i in list_rt:
                code5 = 'delete from t_cust where teacher_id =
"'+str(i[0])+'"'
                code6 = 'delete from t_cust1 where teacher_id =
"'+str(i[0])+'"'
                code7 = 'delete from login where un = "'+i[2]+'"'
                mycur.execute(code5)
                mycur.execute(code6)
                mycur.execute(code7)
                mycon.commit()
                if messagebox.showinfo('Info','Teacher details
successfully removed .'):
```

```
                                    teach_cust()
                        else:
                            teach_cust()
                    entry1['state'] = DISABLED
                    entry2['state'] = DISABLED
                    button12 = Button(twin, text='Delete', font=('Cambria',
15, 'bold'), fg='white', image=image10, height=25, width=100,
compound=CENTER, command=delete_single)
                    button12.place(x=623, y=225)
                else:
                    flag2 += 1
            if flag2 == len(data):
                label12 = Label(twin, text='Not Found', font=('Cambria', 12,
'bold'), fg='red', bg='black')
                label12.place(x=637, y=185)
                twin.after(2500, label12.destroy)
        elif entry1.get() != '' and entry2.get() != '':
            for i in tree2.get_children():
                tree2.delete(i)
            mycur.execute('select * from t_cust')
            data = mycur.fetchall()
```

```
    mycur.execute('select qualification, subject, class_teaching1,
class_teaching2, class_teacher, designation, salary_status from
t_cust1')
    data_t = mycur.fetchall()
    data = list(data)
    for i in range(len(data)):
        for j in range(len(data_t)):
            data[i] = list(data[i])+list(data_t[i])
            break
    flag3 = 0
    for i in data:
        if entry1.get() == i[0] and entry2.get() == i[2]:
            tree2.insert('', END, values=i)
            list_rt.append(i)
            def delete_single_t():
                twin.withdraw()
                m1 = messagebox.askyesno('', 'Are you sure to Delete
?')
                if m1 == True:
                    for i in list_rt:
```

```
                                code5 = 'delete from t_cust where teacher_id =
"'+str(i[0])+'"'
                                code6 = 'delete from t_cust1 where teacher_id =
"'+str(i[0])+'"'
                                code7 = 'delete from login where un = "'+i[2]+'"'
                                mycur.execute(code5)
                                mycur.execute(code6)
                                mycur.execute(code7)
                                mycon.commit()
                                if messagebox.showinfo('Info','Teacher details
successfully removed .'):
                                    teach_cust()
                        else:
                            teach_cust()
                    entry1['state'] = DISABLED
                    entry2['state'] = DISABLED
                    button12 = Button(twin, text='Delete', font=('Cambria',
15, 'bold'), fg='white', image=image10, height=25, width=100,
compound=CENTER, command=delete_single)
                    button12.place(x=623, y=225)
                else:
```

```
                flag3+=1
        if flag3 == len(data):
            label12 = Label(twin, text='Not Found', font=('Cambria', 12, 'bold'), fg='red', bg='black')
            label12.place(x=637, y=185)
            twin.after(2500, label12.destroy)
    elif entry1.get() == '' and entry2.get() == '':
        global a, b
        a += 1
        for i in range(1, 5):
            if a == b or (a, b == 1, 2):
                global label11
                label11 = Label(twin, text='The required information is empty', font=('Cambria', 10, 'bold'), fg='red', bg='black')
                label11.place(x=568, y=185)
                twin.after(2500, label11.destroy)
                break
            else:
                a -= 1
                break
    button11 = Button(twin, text='Show', font=('Cambria', 15, 'bold'), fg='white', image=image10, height=25, width=100, compound=CENTER, command=show_single_t)
    button11.place(x=623, y=225)
def search_teach():
    get_l = []
    l=[]
    mycur.execute('select * from t_cust')
    data2 = mycur.fetchall()
    mycur.execute('select qualification, subject, class_teaching1, class_teaching2, class_teacher, designation, salary_status from t_cust1')
    data_t = mycur.fetchall()
    data2 = list(data2)
    for i in range(len(data2)):
        for j in range(len(data_t)):
            data2[i] = list(data2[i])+list(data_t[i])
            break
    def find():
        f3 = 0
        f4 = 0
        append_l = []
        l_dict = {}
        teaching_dict = {}
        for i in tree2.get_children():
            tree2.delete(i)
        get_l = [entry1t.get().strip(), entry2t.get().strip(), combo1t.get().strip(), combo2t.get().strip(), entry3t.get().strip(), entry4t.get().strip(),combo4t.get().strip(), combo5t.get().strip(), combo3t.get().strip()]
        mycur.execute('select teacher_id,name,gender,blood_group,mother_tongue from t_cust')
        data = mycur.fetchall()
        mycur.execute('select subject,class_teaching1,class_teaching2,class_teacher from t_cust1')
        data_t2 = mycur.fetchall()
        data = list(data)
        for i in range(len(data)):
            for j in range(len(data_t2)):
                data[i] = list(data[i])+list(data_t2[i])
                break
        for i in range(len(data)):
            get_l1 = [data[i][6],data[i][7]]

            if combo4t.get().strip() == '' and combo5t.get().strip() == '':

              get_l = [entry1t.get().strip(), entry2t.get().strip(),
combo1t.get().strip(), combo2t.get().strip(), entry3t.get().strip(),
entry4t.get().strip(),combo4t.get().strip(), combo5t.get().strip(),
combo3t.get().strip()]

        elif combo4t.get().strip() == combo5t.get().strip():

          label9t = Label(twin, text='Enter a valid class.', bg='black',
fg='red', font=('Cambria', 10, 'bold'))

          label9t.place(x=1000, y=250)

          twin.after(1500, label9t.destroy)

        elif combo4t.get().strip() == get_l1[0] or combo5t.get().strip()
== get_l1[1]:

          get_l = [entry1t.get().strip(), entry2t.get().strip(),
combo1t.get().strip(), combo2t.get().strip(), entry3t.get().strip(),
entry4t.get().strip(),combo4t.get().strip(), combo5t.get().strip(),
combo3t.get().strip()]

        elif combo4t.get().strip() == get_l1[1] or combo5t.get().strip()
== get_l1[0]:

          get_l = [entry1t.get().strip(), entry2t.get().strip(),
combo1t.get().strip(), combo2t.get().strip(), entry3t.get().strip(),
entry4t.get().strip(),combo5t.get().strip(), combo4t.get().strip(),
combo3t.get().strip()]

        for i in get_l:
```

```
        if i != '':
            l_dict[get_l.index(i)] = i
            index = get_l.index(i)
            get_l.remove(i)
            get_l.insert(index, '')
    for i in data:
        for j in l_dict:
            if i[j] != None and l_dict[j].lower() == i[j].lower():
                f3 += 1
            if f3 == len(l_dict):
                if data2[data.index(i)] not in append_l:
                    append_l.append(data2[data.index(i)])
                f4 = 1
            if f3 != len(l_dict) and f4 != 1:
                for k in tree2.get_children():
                    tree2.delete(k)
        f3 = 0
for i in append_l:
    tree2.insert('', END, values=i)
    def reset_s():
        entry1t.delete(0, END)
        entry2t.delete(0, END)
        entry3t.delete(0, END)
        entry4t.delete(0, END)
        combo1t.delete(0, END)
        combo2t.delete(0, END)
        combo3t.delete(0, END)
        combo4t.delete(0, END)
        combo5t.delete(0, END)
        for k in tree2.get_children():
            tree2.delete(k)
        for i in data2:
            tree2.insert('', END, values=i)
    button_tc2.destroy()
    button_tc3.destroy()
    button_tc4.destroy()
    button_tc5.destroy()
    label1t = Label(twin, text='Teacher id : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=150, y=50)
    entry1t = ttk.Entry(twin, font=('Cambria', 12, 'bold'))
    entry1t.place(x=330, y=50)
    label2t = Label(twin, text='Name : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=550, y=50)
    entry2t = ttk.Entry(twin, font=('Cambria', 12, 'bold'))
    entry2t.place(x=700, y=50)
    label3t = Label(twin, text='Gender : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=900, y=50)
    list1t = ['Male', 'Female', 'Others']
    combo1t = ttk.Combobox(twin, value=list1t, font=('Cambria', 12, 'bold'))
    combo1t.place(x=1045, y=50)
    label4t = Label(twin, text='Blood Gruop : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=150, y=120)
    list2t = ['O+ve', 'O-ve', 'A+ve', 'A-ve', 'B+ve', 'B-ve', 'AB+ve', 'AB-ve', 'A1+ve', 'Hh']
    combo2t = ttk.Combobox(twin, value=list2t, font=('Cambria', 12, 'bold'))
    combo2t.place(x=330, y=120)
    label5t = Label(twin, text='Mother Tongue : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=550, y=120)
    entry3t = ttk.Entry(twin, font=('Cambria', 12, 'bold'))
    entry3t.place(x=700, y=120)
    label6t = Label(twin, text='Subject : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=900, y=120)
    entry4t = ttk.Entry(twin, font=('Cambria', 12, 'bold'))
    entry4t.place(x=1045, y=120)
    label7t = Label(twin, text='Class Teacher Of : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=270, y=200)
    list3t = ['I', 'II', 'III', 'IV', 'V', 'VI', 'VII', 'VIII', 'IX', 'X', 'XI', 'XII']
    combo3t = ttk.Combobox(twin, value=list3t, font=('Cambria', 12, 'bold'))
    combo3t.place(x=430, y=200)
    label8t = Label(twin, text='Class Teaching : ', font=('Cambria', 12, 'bold'), bg='black', fg='white').place(x=800, y=200)
    list4t = ['I', 'II', 'III', 'IV', 'V', 'VI', 'VII', 'VIII', 'IX', 'X', 'XI', 'XII']
    combo4t = ttk.Combobox(twin, value=list4t, font=('Cambria', 12, 'bold'), width=5, height=8)
    combo4t.place(x=950, y=200)
    combo5t = ttk.Combobox(twin, value=list4t, font=('Cambria', 12, 'bold'), width=5, height=8)
    combo5t.place(x=1100, y=200)
    image9 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg20.png', master=twin)
    image9 = image9.subsample(2, 2)
    button1t = Button(twin, text='Search', font=('Cambria', 12, 'bold'), image=image9, height=25, width=100, bg='black', fg='white', compound=CENTER, command=find)
    button1t.place(x=500, y=250)
    button1t.image = image9
    button2t = Button(twin, text='Reset', font=('Cambria', 12, 'bold'), image=image9, height=25, width=100, bg='black', fg='white', compound=CENTER, command=reset_s)
    button2t.place(x=750, y=250)
    button2t.image = image9


def update_teach():
    get_l2 = []
    search_teach()
    def item_selected(event):
        for selected_item in tree2.selection():
            item = tree2.item(selected_item)
            record = item['values']
        twin.withdraw()
        def replace():
```

```
d_entry1.delete(0, END)
d_entry1.insert(0, record[1])
entry2.insert(0, record[2])
d_entry2.delete(0, END)
d_entry2.insert(0, record[3])
if record[4] == 'male':
    checkbutton1.select()
elif record[4] == 'female':
    checkbutton2.select()
elif record[4] == 'others':
    checkbutton3.select()
combo1.insert(0, record[5])
entry5.insert(0, record[6])
entry11.insert(0, record[7])
entry12.insert(0, record[8])
entry13.insert(0, record[9])
combo2.insert(0, record[10])
combo3.insert(0, record[11])
combo4.insert(0, record[12])
entry14.insert(0, record[13])
```

```
            entry15.insert(0, record[14])
        def save():
            in_data_t = list()
            in_values_t1 = list()
            in_values_t2 = list()
            login_data = list()
            login_values = list()
            in_data_t.append(record[0])
            in_values_t1.append('teacher_id')
            in_values_t2.append('teacher_id')
            in_data_t.append(d_entry1.get())
            in_values_t1.append('joining_date')
            image9 = PhotoImage(file=r'C:\Users\User\Desktop\New
folder\d1.png', master=addt_wn)
            image10 = PhotoImage(file=r'C:\Users\User\Desktop\New
folder\d2.png', master=addt_wn)
            image11 = PhotoImage(file=r'C:\Users\User\Desktop\New
folder\d3.png', master=addt_wn)
            image12 = PhotoImage(file=r'C:\Users\User\Desktop\New
folder\d4.png', master=addt_wn)
```

```
        image13 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d5.png', master=addt_wn)

        image14 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d6.png', master=addt_wn)

        image15 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d7.png', master=addt_wn)

        image16 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d8.png', master=addt_wn)

        image17 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d9.png', master=addt_wn)

        image18 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\d10.png', master=addt_wn)

        if entry2.get().strip() != '':

            in_data_t.append(entry2.get().strip().title())

            in_values_t1.append('name')

            login_values.append('un')

            login_data.append(entry2.get().strip().title())

            login_values.append('up')

            login_data.append(entry2.get().strip().title().replace(' ','')+record[0])

        else:
            label24 = Label(addt_wn, text='●', font=('Cambria', 14, 'bold'),  fg='red', height=20, width=10, image=image9, compound=CENTER)
            label24.image = image9
            label24.place(x=555, y=187)
            addt_wn.after(2500, label24.destroy)
        in_data_t.append(d_entry2.get())
        in_values_t1.append('dob')
        if q == 0:
            gen = record[4]
        in_data_t.append(gen)
        in_values_t1.append('gender')
        if combo1.get().strip() != '':
            in_data_t.append(combo1.get())
            in_values_t1.append('blood_group')
        else:
            label26 = Label(addt_wn, text='●', font=('Cambria', 14, 'bold'), fg='red', height=20, width=10, image=image10, compound=CENTER)
            label26.image = image10
            label26.place(x=570, y=446)
            addt_wn.after(2500, label26.destroy)
        if entry5.get().strip() != '':
            in_data_t.append(entry5.get())
            in_values_t1.append('mother_tongue')
        else:
            label27 = Label(addt_wn, text='●', font=('Cambria', 14,
    'bold'), fg='red', height=20, width=10, image=image11,
    compound=CENTER)
            label27.image = image11
            label27.place(x=553, y=512)
            addt_wn.after(2500, label27.destroy)
        if entry11.get() == '':
            label33 = Label(addt_wn, text='●', font=('Cambria', 14,
    'bold'), fg='red', height=20, width=10, image=image18,
    compound=CENTER)
            label33.image = image18
            label33.place(x=1315, y=255)
            addt_wn.after(2500, label33.destroy)
        else:
            if entry11.get().strip() != '':
```

```
                    if entry11.get().strip().isdigit() and
len(entry11.get().strip()) == 10:
                        in_data_t.append(entry11.get().strip())
                        in_values_t1.append('phone_no')
                    else:
                        label36 = Label(addt_wn, text='Invalid Phone number !
', font=('Cambria', 12, 'bold'), fg='red', bg='black')
                        label36.place(x=1080, y=95)
                        addt_wn.after(2500, label36.destroy)
            if entry12.get().strip() != '':
                in_data_t.append(entry12.get())
                in_values_t2.append('qualification')
            else:
                label35 = Label(addt_wn, text='●', font=('Cambria', 14,
'bold'), fg='red', height=20, width=10, image=image12,
compound=CENTER)
                label35.image = image12
                label35.place(x=1315, y=320)
                addt_wn.after(2500, label35.destroy)
            if entry13.get().strip() != '':
                in_data_t.append(entry13.get().strip())
                in_values_t2.append('subject')
            if combo2.get().strip() != '':
                in_data_t.append(combo2.get().strip())
                in_values_t2.append('class_teaching1')
            if combo3.get().strip() != '':
                in_data_t.append(combo3.get().strip())
                in_values_t2.append('class_teaching2')
            if combo4.get().strip() != '':
                in_data_t.append(combo4.get().strip())
                in_values_t2.append('class_teacher')
                login_values.append('class')
                login_data.append(combo4.get().strip())
            if entry14.get().strip() != '':
                in_data_t.append(entry14.get().strip())
                in_values_t2.append('designation')
                login_values.append('ut')
                login_data.append(entry14.get().strip())
            else:
                label38 = Label(addt_wn, text='●', font=('Cambria', 14,
'bold'), fg='red', height=20, width=10, image=image12,
compound=CENTER)
```

```
            label38.image = image12
            label38.place(x=1315, y=320)
            addt_wn.after(2500, label38.destroy)
        if entry15.get().strip() != '':
            in_data_t.append(entry15.get().strip())
            in_values_t2.append('salary_status')
        else:
            label37 = Label(addt_wn, text='●', font=('Cambria', 14,
'bold'), fg='red', height=20, width=10, image=image12,
compound=CENTER)
            label37.image = image12
            label37.place(x=1315, y=320)
            addt_wn.after(2500, label37.destroy)
        in_values_t2.append('up')
        in_data_t1 = in_data_t[:8]
        in_data_t2 = [in_data_t[0]]+in_data_t[8:]
        in_data_t2.append(entry2.get().strip().title().replace(' ','')+a)
        if len(in_values_t1) != 0 and len(in_data_t1) != 0 and
len(in_values_t2) != 0 and len(in_data_t2) != 0 and len(login_values)
!= 0 and len(login_data) != 0:
        if len(in_values_t1) == len(in_data_t1) and len(in_values_t2)
== len(in_data_t2) and len(login_values) == len(login_data) :
            code0 = 'delete from t_cust where teacher_id =
"'+record[0]+'"'
            code = 'delete from t_cust1 where teacher_id =
"'+record[0]+'"'
            code1 = 'delete from login where un = "'+record[2]+'"'
            code2 = 'insert into t_cust
'+str(tuple(in_values_t1)).replace("'",'')+' values
'+str(tuple(in_data_t1))
            code3 = 'insert into t_cust1
'+str(tuple(in_values_t2)).replace("'",'')+' values
'+str(tuple(in_data_t2))
            code4 = 'insert into login
'+str(tuple(login_values)).replace("'",'')+' values
'+str(tuple(login_data))
            mycur.execute(code0)
            mycur.execute(code)
            mycur.execute(code1)
            mycur.execute(code2)
            mycur.execute(code3)
            mycur.execute(code4)
            mycur.execute('commit')
```

```
          addt_wn.withdraw()
          if messagebox.showinfo('Info','Teacher details
successfully saved .'):
            teach_cust()
    def reset():
      d_entry1.delete(0, END)
      d_entry1.insert(0, date1)
      entry2.delete(0, END)
      entry2.insert(0, '')
      d_entry2.delete(0, END)
      dob2 = str(date.today())[8:] + '/' + str(date.today())[5:7] + '/'
+ str(int(str(date.today())[:4]) - 20)
      d_entry2.insert(0, dob2)
      checkbutton1.deselect()
      checkbutton2.deselect()
      checkbutton3.deselect()
      combo1.delete(0, END)
      combo1.insert(0, '')
      combo2.delete(0, END)
      combo2.insert(0, '')
      combo3.delete(0, END)
      combo3.insert(0, '')
      combo4.delete(0, END)
      combo4.insert(0, '')
      entry5.delete(0, END)
      entry5.insert(0, '')
      entry11.delete(0, END)
      entry11.insert(0, '')
      entry12.delete(0, END)
      entry12.insert(0, '')
      entry13.delete(0, END)
      entry13.insert(0, '')
      entry14.delete(0, END)
      entry14.insert(0, '')
      entry15.delete(0, END)
      entry15.insert(0, '')
    def cancel():
      addt_wn.withdraw()
      if messagebox.askyesno('', 'Are you sure to Exit ?'):
          teach_cust()


    global image9
    twin.withdraw()
    addt_wn = Tk()
```

```
    addt_wn.title('Append students data')
    addt_wn.state('zoomed')
    icon12 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\office5.png', master=addt_wn)
    addt_wn.iconphoto(False, icon12)
    image7 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg18.png', master=addt_wn)
    label23 = Label(addt_wn, image=image7).place(x=-2, y=-2)
    label3 = Label(addt_wn, text='APPEND DATA', font=('Cambria', 20), bg='black', fg='gray').pack()
    label4 = Label(addt_wn, text='Teacher Id : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=50, y=57)
    label5 = Label(addt_wn, text=record[0], font=('Cambria', 15, 'bold'), bg='black', fg='cyan').place(x=300, y=57)
    label6 = Label(addt_wn, text='Joining Date : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=50, y=137)
    d_entry1 = DateEntry(addt_wn, font=('Cambria', 15, 'bold'), width=20, selectmode='day', maxdate=date.today(), background='black', foreground='white', date_pattern='dd/mm/y')
    d_entry1.place(x=300, y=137)
    date1 = d_entry1.get()
    dob = date1.replace(date1[6:], str(int(date1[6:])-20))
    dob = date(int(dob[6:]), int(dob[3:5]), int(dob[:2]))

    d_entry1.focus()

    label7 = Label(addt_wn, text='Name of the Teacher : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=50, y=217)

    entry2 = ttk.Entry(addt_wn, font=('Cambria', 15, 'bold'))

    entry2.place(x=300, y=217)

    label9 = Label(addt_wn, text='Date of Birth : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=50, y=297)

    d_entry2 = DateEntry(addt_wn, font=('Cambria', 15, 'bold'), width=20, selectmode='day', maxdate=dob, background='black', foreground='white', date_pattern='dd/mm/y')

    d_entry2.place(x=300, y=297)

    label10 = Label(addt_wn, text='Gender : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=50, y=377)

    def male():

        global gen, q

        q = 1

        gen = 'male'

    def female():

        global gen, q

        q = 2
        gen = 'female'

    def others():

        global gen, q

        q = 3

        gen = 'others'

    checkbutton1 = Checkbutton(addt_wn, text='Male',
bg='lightblue', activebackground='lightblue', height=1, variable=1,
onvalue=1, offvalue=0, font=('Cambria', 12, 'bold'), command=male)

    checkbutton1.place(x=300, y=377)

    checkbutton2 = Checkbutton(addt_wn, text='Female',
bg='lightblue', activebackground='lightblue', height=1, variable=1,
onvalue=2, offvalue=0, font=('Cambria', 12, 'bold'),
command=female)

    checkbutton2.place(x=400, y=377)

    checkbutton3 = Checkbutton(addt_wn, text='Others',
bg='lightblue', activebackground='lightblue', height=1, variable=1,
onvalue=3, offvalue=0, font=('Cambria', 12, 'bold'),
command=others)

    checkbutton3.place(x=525, y=377)

    label11 = Label(addt_wn, text='Blood Group : ', font=('Cambria',
15, 'bold'), bg='black', fg='white').place(x=50, y=457)

    list1 = ['O+ve', 'O-ve', 'A+ve', 'A-ve', 'B+ve', 'B-ve', 'AB+ve', 'AB-
ve', 'A1+ve', 'Hh']
```

```
    combo1 = ttk.Combobox(addt_wn, value=list1, font=('Cambria', 15, 'bold'))

    combo1.place(x=300, y=457)

    label12 = Label(addt_wn, text='Mother Tongue : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=50, y=537)

    entry5 = ttk.Entry(addt_wn, font=('Cambria', 15, 'bold'))

    entry5.place(x=300, y=537)

    label18 = Label(addt_wn, text="Phone Number : ", font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=800, y=57)

    entry11 = ttk.Entry(addt_wn, font=('Cambria', 15, 'bold'))

    entry11.place(x=1050, y=57)

    label19 = Label(addt_wn, text="Qualification : ", font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=800, y=137)

    entry12 = ttk.Entry(addt_wn, font=('Cambria', 15, 'bold'))

    entry12.place(x=1050, y=137)

    label20 = Label(addt_wn, text='Subject : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=800, y=217)

    entry13 = ttk.Entry(addt_wn, font=('Cambria', 15, 'bold'))

    entry13.place(x=1050, y=217)
    label20 = Label(addt_wn, text='Class Teaching : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=800, y=297)

    list3 = ['None','I', 'II', 'III', 'IV', 'V', 'VI', 'VII', 'VIII', 'IX', 'X', 'XI', 'XII']

    combo2 = ttk.Combobox(addt_wn, value=list3, font=('Cambria', 15, 'bold'), width=5, height=10)

    combo2.place(x=1050, y=297)

    combo3 = ttk.Combobox(addt_wn, value=list3, font=('Cambria', 15, 'bold'), width=5, height=10)

    combo3.place(x=1200, y=297)

    label20 = Label(addt_wn, text='Class Teacher : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=800, y=377)

    list4 = ['I', 'II', 'III', 'IV', 'V', 'VI', 'VII', 'VIII', 'IX', 'X', 'XI', 'XII']

    combo4 = ttk.Combobox(addt_wn, value=list4, font=('Cambria', 15, 'bold'))

    combo4.place(x=1050, y=377)

    label20 = Label(addt_wn, text='Designation : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=800, y=457)

    entry14 = ttk.Entry(addt_wn, font=('Cambria', 15, 'bold'))

    entry14.place(x=1050, y=457)
```

```
        label20 = Label(addt_wn, text='Salary Status : ', font=('Cambria', 15, 'bold'), bg='black', fg='white').place(x=800, y=537)

        entry15 = ttk.Entry(addt_wn, font=('Cambria', 15, 'bold'))

        entry15.place(x=1050, y=537)

        image9 = PhotoImage(file=r'C:\Users\User\Desktop\New folder\bg20.png', master=addt_wn)

        image9 = image9.subsample(2, 2)

        button13 = Button(addt_wn, text='Save', font=('Cambria', 15, 'bold'), image=image9, fg='white', height=30, width=120, command=save, compound=CENTER)

        button13.place(x=800, y=660)

        button14 = Button(addt_wn, text='Reset', font=('Cambria', 15, 'bold'), image=image9, fg='white', height=30, width=120, command=reset, compound=CENTER)

        button14.place(x=1000, y=660)

        button15 = Button(addt_wn, text='Cancel', font=('Cambria', 15, 'bold'), image=image9, fg='white', height=30, width=120, command=cancel, compound=CENTER)

        button15.place(x=1200, y=660)

        replace()

        addt_wn.mainloop()

    tree2.bind('<Double-1>', item_selected)

office_func()
```

FUTURE ENHANCEMENTS

Since this project is just a skeleton for school management, many enhancements can be made in this project.

- Adding some more tkinter widgets, which gives a complete look for a school management system.
- A new module can be added where the admin will have an option to check the login time of all users.
- Billing system can be add to this project.
- This can be made more user friendly, so that users can have an easy access.

BIBLIOGRAPHY

- Computer Science with Python Class 12 BY Sumita Arora
- Computer Science with Python Class 11 BY Sumita Arora
- Computer Science with Python Class 12 BY Preeti Arora
- Computer Science with Python Class 11 BY Preeti Arora
- Stackoverflow.com
- www.w3schools.com
- www.geeksforgeeks.org