In [110…

```
from tkinter import *
import tkinter as tk
from tkinter import ttk
from tkinter import messagebox
from PIL import ImageTk , Image
from tkinter import messagebox
import sqlite3


win = Tk()
win.title("ERP Login")
win.geometry("925x500+300+200")
win.configure(bg = "white")
win.resizable(0 , 0)

#////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#Sign_In Function
def Sign_In():
    username = user_username.get()
    password = user_password.get()

    if username == 'Owner_Login' and password == 'Get_Access' :

        root = Toplevel(win)
        root.title("ERP")
        root.geometry("1225x700+150+50")
        root.configure(bg = "white")
        root.resizable(0 , 0)

#////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#Frames
        title_Label = tk.Label(root , text = "Student Management System" , font = ("Times Nre Roman" , 30 , "bold") , bd = 12
        title_Label.pack(side = TOP , fill = X)

        Detial_Frame = tk.LabelFrame(root , bd = 12 , relief = GROOVE , width = 450 , height = 639 , bg = "lightgrey")
        Detial_Frame.place(x = 0 , y = 65)
```

```
        Detial_Label = tk.Label(root , text = "Enter Detials" , font = ("Times New Roman" , 25 , "bold"), bg = "lightgrey")
        Detial_Label.place(x = 135 , y = 80)

        DataBase_Frame = tk.Frame(root , bd = 12 , relief = GROOVE , width = 777 , height = 639 , bg = "lightgrey")
        DataBase_Frame.place(x = 447 , y = 65)

#////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#Detials_Entry
        Name_Label = tk.Label(Detial_Frame , text = "Name" , font = ("Times New Roman" , 22 , "bold"), bg = "lightgrey")
        Name_Label.place(x = 7 , y = 60)
        Name_Label_Entry = tk.Entry(Detial_Frame , bd = 3 , width = 45 )
        Name_Label_Entry.place(x = 117 , y = 70)

        Email_Label = tk.Label(Detial_Frame , text = "Email" , font = ("Times New Roman" , 22 , "bold"), bg = "lightgrey")
        Email_Label.place(x = 7 , y = 100)
        Email_Label_Entry = tk.Entry(Detial_Frame , bd = 3 , width = 45  )
        Email_Label_Entry.place(x = 117 , y = 110)

        Contact_Label = tk.Label(Detial_Frame , text = "Contact" , font = ("Times New Roman" , 22 , "bold"), bg = "lightgrey")
        Contact_Label.place(x = 7 , y = 140)
        Contact_Label_Entry = tk.Entry(Detial_Frame , bd = 3 , width = 45 )
        Contact_Label_Entry.place(x = 117 , y = 150)

        Class_Label = tk.Label(Detial_Frame , text = "Class" , font = ("Times New Roman" , 22 , "bold"), bg = "lightgrey")
        Class_Label.place(x = 7 , y = 180)
        Class_Label_Entry = tk.Entry(Detial_Frame , bd = 3 , width = 45 )
        Class_Label_Entry.place(x = 117 , y = 190)

        Batch_Label = tk.Label(Detial_Frame , text = "Batch" , font = ("Times New Roman" , 22 , "bold"), bg = "lightgrey")
        Batch_Label.place(x = 7 , y = 220)
        Batch_Label_Entry = tk.Entry(Detial_Frame , bd = 3 , width = 45 )
        Batch_Label_Entry.place(x = 117 , y = 230)

        Prn_Label = tk.Label(Detial_Frame , text = "PRN" , font = ("Times New Roman" , 22 , "bold"), bg = "lightgrey")
        Prn_Label.place(x = 7 , y = 260)
        Prn_Label_Entry = tk.Entry(Detial_Frame , bd = 3 , width = 45 )
        Prn_Label_Entry.place(x = 117 , y = 270)

        Gender_Label = tk.Label(Detial_Frame , text = "Gender" , font = ("Times New Roman" , 22 , "bold"), bg = "lightgrey")
        Gender_Label.place(x = 7 , y = 300)
```

```
        Gender_Label_Entry = ttk.Combobox(Detial_Frame ,font = ("Times New Roman" , 12) , width = 32)
        Gender_Label_Entry['values'] = ("Male" , "Female" , "Others")
        Gender_Label_Entry.place(x = 117 , y = 310)

        Address_Label = tk.Label(Detial_Frame , text = "Address" , font = ("Times New Roman" , 22 , "bold"), bg = "lightgrey")
        Address_Label.place(x = 7 , y = 340)
        Address_Label_Entry = tk.Entry(Detial_Frame , bd = 3 , width = 45 )
        Address_Label_Entry.place(x = 117 , y = 350)

        Dob_Label = tk.Label(Detial_Frame , text = "DOB" , font = ("Times New Roman" , 22 , "bold"), bg = "lightgrey")
        Dob_Label.place(x = 7 , y = 380)
        Dob_Label_Entry = tk.Entry(Detial_Frame , bd = 3 , width = 45 )
        Dob_Label_Entry.place(x = 117 , y = 390)

#////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#Create DataBase
        connection = sqlite3.connect("APP_Final_Project_Submission.db")
        cursor = connection.cursor()
        print("Connection Is Established")

        cursor.execute(''' CREATE TABLE IF NOT EXISTS students(Id INTEGER PRIMARY KEY AUTOINCREMENT , Name TEXT , Email TEXT ,
        connection.commit()


#Function To Add Data
        def Add_Data():
            Name = Name_Label_Entry.get()
            Email = Email_Label_Entry.get()
            Contact = Contact_Label_Entry.get()
            Class = Class_Label_Entry.get()
            Batch = Batch_Label_Entry.get()
            Prn = Prn_Label_Entry.get()
            Gender = Gender_Label_Entry.get()
            Address = Address_Label_Entry.get()
            Dob = Dob_Label_Entry.get()

            if Name and Email and Contact and Class and Batch and Prn and Gender and Address and Dob :

                cursor.execute("INSERT INTO students (Name , Email , Contact , Class , Batch , Prn , Gender , Address , Dob) V
                connection.commit()
```

```
                Update_Treeview()

            else:
                messagebox.showerror('Invalid' , 'Please Fill Out All Fileds')


#Function To Display Data
        def Update_Treeview():
            for row in Data_Table.get_children():
                Data_Table.delete(row)

            cursor.execute('SELECT * FROM students')
            data = cursor.fetchall()

            for student in data:
                Data_Table.insert('', 'end', values=student)


#Function To Delete Data
        def Delete_Data():
            if not Data_Table.selection():
                messagebox.showerror('Invalid' , 'Please Select An Item From The Database')

            else:
                current_item = Data_Table.focus()
                values = Data_Table.item(current_item)
                selection = values["values"]
                Data_Table.delete(current_item)
                connection.execute("DELETE FROM students WHERE Id=%d" % selection[0])
                connection.commit()
                messagebox.showinfo('Successfull', 'The Data Is Deleted Successfully.')
                Update_Treeview()


#Function To Clear Info
        def Clear_Data():
            Name_Label_Entry.delete(0, tk.END)
            Email_Label_Entry.delete(0, tk.END)
            Contact_Label_Entry.delete(0, tk.END)
            Class_Label_Entry.delete(0, tk.END)
            Batch_Label_Entry.delete(0, tk.END)
```

```
            Prn_Label_Entry.delete(0, tk.END)
            Gender_Label_Entry.delete(0, tk.END)
            Address_Label_Entry.delete(0, tk.END)
            Dob_Label_Entry.delete(0, tk.END)

#////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

        Add_Button = tk.Button(Detial_Frame , text = "Add" , font = ("Times New Roman" , 17 , "bold" ), bd = 5 , width = 19, b
        Add_Button.place(x = 80 , y = 455)

        Delete_Button = tk.Button(Detial_Frame , text = "Delete" , font = ("Times New Roman" , 17 , "bold" ), bd = 5 , width =
        Delete_Button.place(x = 65 , y = 525)

        Clear_Button = tk.Button(Detial_Frame , text = "Clear" , font = ("Times New Roman" , 17 , "bold" ), bd = 5 , width = 1
        Clear_Button.place(x = 229 , y = 525)

#////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

        Detials_Frame = tk.Frame(DataBase_Frame , bd = 4 , relief = tk.GROOVE , width = 757 , height = 47 , bg = "lightgrey" )
        Detials_Frame.pack(side = tk.TOP )

        Detials_Label = tk.Label(DataBase_Frame , text = "STUDENT DATA" , font = ("Times New Roman" , 18 , "bold") , bg = "lig
        Detials_Label.place(x = 310 , y = 5)

#////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////


        Main_Frame = tk.Frame(DataBase_Frame , bd = 9 , relief = tk.GROOVE , height = 630 , width = 750 , bg = "lightgrey")
        Main_Frame.pack(fill = tk.BOTH , expand = True)

        Data_Table = ttk.Treeview(Main_Frame , columns = ("Id" , "Name" , "Email" , "Contact" , "Class" , "Batch" , "Prn" , "G
        Data_Table.pack(side = tk.TOP)


        Data_Table.heading("Id" , text = "ID")
        Data_Table.heading("Name" , text = "Name")
        Data_Table.heading("Email" , text = "Email")
        Data_Table.heading("Contact" , text = "Contact")
        Data_Table.heading("Class" , text = "Class")
        Data_Table.heading("Batch" , text = "Batch")
        Data_Table.heading("Prn" , text = "Prn")
```

```
        Data_Table.heading("Gender" , text = "Gender")
        Data_Table.heading("Address" , text = "Address")
        Data_Table.heading("Dob" , text = "Dob")

        Data_Table['show'] = 'headings'

        Data_Table.column("Id" , width = 40)
        Data_Table.column("Name" , width = 110)
        Data_Table.column("Email" , width = 140)
        Data_Table.column("Contact" , width = 100)
        Data_Table.column("Class" ,width = 50)
        Data_Table.column("Batch" , width = 50)
        Data_Table.column("Prn" , width = 50)
        Data_Table.column("Gender" , width = 50)
        Data_Table.column("Address" , width = 120)
        Data_Table.column("Dob" , width = 40)

        Update_Treeview()

#////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#Result Data
        cursor.execute(''' CREATE TABLE IF NOT EXISTS marks(Id INTEGER PRIMARY KEY AUTOINCREMENT , Name TEXT , Prn INTEGER ,
        connection.commit()



#Function To Add marks
        def Add_Calculate_Marks() :
            Name = Name_Label_Entry.get()
            Prn = int(Prn_Label_Entry.get())
            App = int(APP_Label_Entry.get())
            Lait = int(LAIT_Label_Entry.get())
            Es = int(ES_Label_Entry.get())
            Ds = int(DS_Label_Entry.get())
            Dms = int(DMS_Label_Entry.get())
            Pld = int(PLD_Label_Entry.get())


            Total = (App + Lait + Es + Ds + Dms +  Pld)
            tk.Label(Buttons_Frame , text = f"{Total}" , font = "arial 12 bold" , bg = "lightgrey").place(x = 55 , y = 3)
```

```
            Percentage = int(Total/6)
            Percentage_Label = tk.Label(Buttons_Frame , text = f"{Percentage}%" , font = "arial 12 bold" ,  bg = "lightgrey").

            if (Percentage >= 35) :
                Result = "PASS"

            else :
                Result = "FAIL"

            tk.Label(Buttons_Frame , text = f"{Result}" , font = "arial 10 bold" , bg = "lightgrey").place(x = 153 , y =4)


            if Name and Prn and App and Lait and Es and Ds and Dms and Pld :

                cursor.execute("INSERT INTO marks (Name , Prn , App , Lait , Es , Ds , Dms , Pld) VALUES (? , ? , ? , ? , ? ,
                connection.commit()
                Update_Treeview_Marks()

            else:
                messagebox.showerror('Invalid' , 'Please Fill Out All Fileds')


#Function To Display Data
        def Update_Treeview_Marks():
            for row in Marks_Table.get_children():
                Marks_Table.delete(row)

            cursor.execute('SELECT * FROM marks')
            data = cursor.fetchall()

            for student in data:
                Marks_Table.insert('', 'end', values=student)


#Function To DeletenD ata
        def Delete_Marks_Data():
            if not Marks_Table.selection():
                messagebox.showerror('Invalid' , 'Please Select An Item From The Database')

            else:
```

```
                current_item = Marks_Table.focus()
                values = Marks_Table.item(current_item)
                selection = values["values"]
                Marks_Table.delete(current_item)
                connection.execute("DELETE FROM marks WHERE Id=%d" % selection[0])
                connection.commit()
                messagebox.showinfo('Successfull', 'The Data Is Deleted Successfully.')
                Update_Treeview_Marks()


#Function To Clear Info
        def Clear_Marks_Data():
            Name_Label_Entry.delete(0, tk.END)
            Prn_Label_Entry.delete(0, tk.END)
            APP_Label_Entry.delete(0, tk.END)
            LAIT_Label_Entry.delete(0, tk.END)
            ES_Label_Entry.delete(0, tk.END)
            DS_Label_Entry.delete(0, tk.END)
            DMS_Label_Entry.delete(0, tk.END)
            PLD_Label_Entry.delete(0, tk.END)


#////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////


        Marks_Frame = tk.Frame(DataBase_Frame , bd = 9 , relief = tk.GROOVE , width = 757 , height = 90 , bg = "lightgrey" )
        Marks_Frame.pack(side = tk.BOTTOM)

        APP_Label = tk.Label(Marks_Frame , text = "APP :-" , font = ("Times New Roman" , 12) , bg = "lightgrey")
        APP_Label.place(x = 7 , y  = 7)
        APP_Label_Entry = tk.Entry(Marks_Frame , bd = 3 , width = 5)
        APP_Label_Entry.place(x = 58 , y  = 9)

        LAIT_Label = tk.Label(Marks_Frame , text = "LAIT :-" , font = ("Times New Roman" , 12) , bg = "lightgrey")
        LAIT_Label.place(x = 7 , y  = 40)
        LAIT_Label_Entry = tk.Entry(Marks_Frame , bd = 3 , width = 5)
        LAIT_Label_Entry.place(x = 58 , y  = 42)

        ES_Label = tk.Label(Marks_Frame , text = "ES :-" , font = ("Times New Roman" , 12) , bg = "lightgrey")
        ES_Label.place(x = 107 , y  = 7)
        ES_Label_Entry = tk.Entry(Marks_Frame , bd = 3 , width = 5)
```

```
        ES_Label_Entry.place(x = 150 , y  = 9)


        DS_Label = tk.Label(Marks_Frame , text = "DS :-" , font = ("Times New Roman" , 12) , bg = "lightgrey")
        DS_Label.place(x = 107 , y  = 40)
        DS_Label_Entry = tk.Entry(Marks_Frame , bd = 3 , width = 5)
        DS_Label_Entry.place(x = 150 , y  = 42)

        DMS_Label = tk.Label(Marks_Frame , text = "DMS :-" , font = ("Times New Roman" , 12) , bg = "lightgrey")
        DMS_Label.place(x = 207 , y  = 7)
        DMS_Label_Entry = tk.Entry(Marks_Frame , bd = 3 , width = 5)
        DMS_Label_Entry.place(x = 260 , y  = 9)

        PLD_Label = tk.Label(Marks_Frame , text = "PLD :-" , font = ("Times New Roman" , 12) , bg = "lightgrey")
        PLD_Label.place(x = 207 , y  = 40)
        PLD_Label_Entry = tk.Entry(Marks_Frame , bd = 3 , width = 5)
        PLD_Label_Entry.place(x = 260 , y  = 42)

        Buttons_Frame = tk.Frame(Marks_Frame , bd = 7 , relief  = tk.GROOVE , width = 420 , height = 70 , bg = "lightgrey")
        Buttons_Frame.place(x = 318 , y = 0)

        Total_Label = tk.Label(Buttons_Frame , text = "Total :-" , font = ("Times New Roman" , 12) , bg = "lightgrey" , fg = "
        Total_Label.place(x = 5 , y = 2)

        Percentage_Label = tk.Label(Buttons_Frame , text = "Percentage :-" , font = ("Times New Roman" , 12) , bg = "lightgrey
        Percentage_Label.place(x = 5 , y = 28)

        Result_Label = tk.Label(Buttons_Frame , text = "Result :-" , font = ("Times New Roman" , 12) , bg = "lightgrey" , fg =
        Result_Label.place(x = 100 , y = 2)

        Add_Marks_Button = tk.Button(Buttons_Frame , text = "Add Marks" , font = ("Times New Roman" , 9) , width = 19 , bg = "
        Add_Marks_Button.place(x = 221 , y = 0)

        Clear_Marks_Button = tk.Button(Buttons_Frame , text = "Clear" , font = ("Times New Roman" , 9) , width = 8 , bg = "lig
        Clear_Marks_Button.place(x = 220 , y = 28)

        Delete_Marks_Button = tk.Button(Buttons_Frame , text = "Delete" , font = ("Times New Roman" , 9) , width = 8 , bg = "l
        Delete_Marks_Button.place(x = 300 , y = 28)

#////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
        Marks_Frame = tk.Frame(DataBase_Frame , bd = 4 , relief = tk.GROOVE , bg  = "lightgrey" , width = 747 , height = 285)
        Marks_Frame.pack(side = tk.TOP)

        Marks_Table = ttk.Treeview(Marks_Frame , columns = ("Id" , "Name" , "PRN" , "APP" , "DS" , "LAIT" , "DMS" , "ES" , "PL
        Marks_Table.pack(side = tk.TOP)

        Marks_Table.heading("Id" , text = "ID")
        Marks_Table.heading("Name" , text = "Name")
        Marks_Table.heading("PRN" , text = "PRN")
        Marks_Table.heading("APP" , text = "APP")
        Marks_Table.heading("DS" , text = "DS")
        Marks_Table.heading("LAIT" , text = "LAIT")
        Marks_Table.heading("DMS" , text = "DMS")
        Marks_Table.heading("ES" , text = "ES")
        Marks_Table.heading("PLD" , text = "PLD")
        Marks_Table.heading("PERCENTAGE" , text = "Percentage")
        Marks_Table.heading("RESULT" , text = "Result")

        Marks_Table['show'] = 'headings'

        Marks_Table.column("Id" , width = 45)
        Marks_Table.column("Name" , width = 140)
        Marks_Table.column("PRN" , width = 100)
        Marks_Table.column("APP" , width = 50)
        Marks_Table.column("DS" ,width = 50)
        Marks_Table.column("LAIT" , width = 50)
        Marks_Table.column("DMS" , width = 50)
        Marks_Table.column("ES" , width = 60)
        Marks_Table.column("PLD" , width = 60)
        Marks_Table.column("PERCENTAGE" , width = 70)
        Marks_Table.column("RESULT" , width = 70)

#////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

    elif username != 'Teacher' and password != 'Mini_Project' :
        messagebox.showerror('Invalid' , 'INVALID USERNAME & PASSWORD')

    elif password != 'Mini_Project' :
        messagebox.showerror('Invalid' , 'INCORRECT PASSWORD')

    elif username != 'Teacher' :
```

```
        messagebox.showerror('Invalid' , 'INVALID USERNAME')

        root.mainloop()
        connection.close()

#////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

# Image PART
img = Image.open("C:/Users/sarth/Downloads/login.png")
img_test = ImageTk.PhotoImage(img)
label1 = Label(win , image = img_test , bg = "white" )
label1.image = img_test
label1.place(x = 50 , y = 50)

# Signin_Frame
frame = Frame(win , width = 350 , height = 350 , bg = "white")
frame.place(x = 480 , y = 70)

signin_heading = Label(frame , text = "Sign In" , font = ("Microsoft YaHei UI Light" , 23 , "bold") ,fg = "#6495ED" , bg = "wh
signin_heading.place(x = 120 , y = 5)

#Username
def on_click(e) :
    user_username.delete(0 , 'end')

def on_release(e) :
    name = user_username.get()
    if name == '' :
        user_username.insert(0 , 'Username')

user_username = Entry(frame , width = 33 ,fg = "black" , bg = "white" , font = ("Microsoft YaHei UI Light" , 12 ) , bd = 0)
user_username.place(x= 30 , y = 80)
user_username.insert(0 , "Username")
user_username.bind('<FocusIn>' , on_click)
user_username.bind('<FocusOut>' , on_release)

frame1 = Frame(frame , width = 295 , height = 2 , bg = "black")
frame1.place(x = 27 , y = 107)

#Password
def on_click(e) :
```

```
    user_password.delete(0 , 'end')

def on_release(e) :
    name = user_password.get()
    if name == '' :
        user_password.insert(0 , 'Password')

user_password = Entry(frame , width = 33 ,fg = "black" , bg = "white" , font = ("Microsoft YaHei UI Light" , 12  ) , bd = 0)
user_password.place(x= 28 , y = 150)
user_password.insert(0 , "Password")
user_password.bind('<FocusIn>' , on_click)
user_password.bind('<FocusOut>' , on_release)

frame2 = Frame(frame , width = 295 , height = 2 , bg = "black")
frame2.place(x = 27 , y = 177)

#SignIn Button
Sign_In = Button(frame , text = "Sign In" , font = ("Microsoft YaHei UI Light" , 11) , width = 30 , height = 1 , bg = "#6495ED
Sign_In.place(x = 35 , y = 210)

win.mainloop()
```

```
Connection Is Established
```