

Source Code Of RMS Project

Report.py

```
from tkinter import*
from PIL import Image,ImageTk #pip install pillow
from tkinter import ttk,messagebox
import sqlite3
class reportClass:
    def __init__(self,root):
        self.root=root
        self.root.title("Student Result Management System")
        self.root.geometry("1200x480+80+170")
        self.root.config(bg="white")
        self.root.focus_force()
        #=====title=====
        title=Label(self.root,text="View Student Result",font=("goudy old
style",20,"bold"),bg="orange",fg="#262626").place(x=10,y=15,width=1180,height=50)
        #=====search=====
        self.var_search=StringVar()
        self.var_id=""
        lbl_search=Label(self.root,text="Search by roll no.", font=("goudy old
style",20,"bold"),bg="white").place(x=280,y=100)
        txt_search=Entry(self.root,textvariable=self.var_search, font=("goudy old
style",20),bg="lightyellow").place(x=520,y=100,width=150)
        btn_search=Button(self.root,text='Search',font=("goudy old
style",15,"bold"),bg="#03a9f4",fg="white",cursor="hand2",command=self.search).plac
e(x=680,y=100,width=100,height=35)
        btn_clear=Button(self.root,text='clear',font=("goudy old
style",15,"bold"),bg="gray",fg="white",cursor="hand2",command=self.clear).place(x=80
0,y=100,width=100,height=35)
        #=====results_label=====
```

```
lbl_roll=Label(self.root,text="Roll No",font=("goudy old  
style",15,"bold"),bg="white",bd=2,relief=GROOVE).place(x=150,y=230,width=150,height=5  
0)
```

```
lbl_name=Label(self.root,text="Name",font=("goudy old  
style",15,'bold'),bg="white",bd=2,relief=GROOVE).place(x=300,y=230,width=150,height=5  
0)
```

```
lbl_course=Label(self.root,text="Course",font=("goudy old  
style",15,'bold'),bg="white",bd=2,relief=GROOVE).place(x=450,y=230,width=150,height=5  
0)
```

```
lbl_marks=Label(self.root,text="Marks Obtained",font=("goudy old  
style",15,'bold'),bg="white",bd=2,relief=GROOVE).place(x=600,y=230,width=150,height=5  
0)
```

```
lbl_full=Label(self.root,text="Total Marks",font=("goudy old  
style",15,'bold'),bg="white",bd=2,relief=GROOVE).place(x=750,y=230,width=150,height=5  
0)
```

```
lbl_per=Label(self.root,text="Percentage",font=("goudy old  
style",15,'bold'),bg="white",bd=2,relief=GROOVE).place(x=900,y=230,width=150,height=5  
0)
```

```
self.roll=Label(self.root,font=("goudy old  
style",15,"bold"),bg="white",bd=2,relief=GROOVE)
```

```
self.roll.place(x=150,y=280,width=150,height=50)
```

```
self.name=Label(self.root,font=("goudy old  
style",15,'bold'),bg="white",bd=2,relief=GROOVE)
```

```
self.name.place(x=300,y=280,width=150,height=50)
```

```
self.course=Label(self.root,font=("goudy old  
style",15,'bold'),bg="white",bd=2,relief=GROOVE)
```

```
self.course.place(x=450,y=280,width=150,height=50)
```

```
self.marks=Label(self.root,font=("goudy old  
style",15,'bold'),bg="white",bd=2,relief=GROOVE)
```

```
self.marks.place(x=600,y=280,width=150,height=50)
```

```
self.full=Label(self.root,font=("goudy old  
style",15,'bold'),bg="white",bd=2,relief=GROOVE)
```

```
self.full.place(x=750,y=280,width=150,height=50)
```

```
self.per=Label(self.root,font=("goudy old  
style",15,'bold'),bg="white",bd=2,relief=GROOVE)
```

```

self.per.place(x=900,y=280,width=150,height=50)

#=====button delete=====

btn_delete=Button(self.root,text='Delete',font=("goudy old
style",15,"bold"),bg="red",fg="white",cursor="hand2",command=self.delete).place(x=50
0,y=350,width=150,height=35)

#=====
=====

def search(self):
    con=sqlite3.connect(database="rms.db")
    cur=con.cursor()
    try:
        if self.var_search.get()=="":
            messagebox.showerror("Error","Roll no should be
required",parent=self.root)
        else:
            cur.execute(f"select * from result where roll=?", (self.var_search.get(),))
            row=cur.fetchone()
            if row!=None:
                self.var_id=row[0]
                self.roll.config(text=row[1])
                self.name.config(text=row[2])
                self.course.config(text=row[3])
                self.marks.config(text=row[4])
                self.full.config(text=row[5])
                self.per.config(text=row[6])
            else:
                messagebox.showerror("Error","No record found",parent=self.root)

```

```

except Exception as ex:
    messagebox.showerror("Error",f"Error due to {str(ex)}")

def clear(self):
    self.var_id=""
    self.roll.config(text="")
    self.name.config(text="")
    self.course.config(text="")
    self.marks.config(text="")
    self.full.config(text="")
    self.per.config(text="")
    self.var_search.set("")

def delete(self):
    con=sqlite3.connect(database="rms.db")
    cur=con.cursor()
    try:
        if self.var_id=="":
            messagebox.showerror("Error","Search Student result
first",parent=self.root)
        else:
            cur.execute("select * from result where rid=?", (self.var_id,))
            row=cur.fetchone()
            print(row)
            if row==None:
                messagebox.showerror("Error","Invaild Student result
",parent=self.root)
            else:

```

```
        op=messagebox.askyesno("Confirm","Do you really want to
delete?",parent=self.root)

        if op==True:
            cur.execute("delete from result where rid=?", (self.var_id,))
            con.commit()

            messagebox.showinfo("Delete","Result deleted
Successfully",parent=self.root)

            self.clear()

    except Exception as ex:
        messagebox.showerror("Error",f"Error due to {str(ex)}")

if __name__=="__main__":
    root=Tk()
    obj=reportClass(root)
    root.mainloop()
```