```
import tkinter as tk
from tkinter import messagebox

import sqlite3 as sql

class AttendanceManager(tk.Tk):
def __init__(self,*args,**kwargs):

tk.Tk. __init__(self,*args,**kwargs) container=tk.Frame(self)

container.pack(side="top",fill="both",expand=True) container.grid_rowconfigure(0,weight=1)
container.grid_columnconfigure(0,weight=1)

self.frames=dict()

for F in
(StartPage,NewRecord,ManageAttendance,DeleteRecord,EditRecord,AddSubjects,TodayDa
ta):

frame=F(container,self) self.frames[F]=frame frame.grid(row=0,column=0,sticky="nsew")

self.show_frame(StartPage) def show_frame(self,cont):

frame=self.frames[cont]

frame.tkraise()

class StartPage(tk.Frame):
def __init__(self,parent,controller):

tk.Frame.__init__(self,parent)

label1=tk.Label(self,text="Hi fella!, what do you desire?",font=("Times",24))

bt1=tk.Button(self,text="Start a new
record",font=("Times",16),height=2,width=17,command=lambda:controller.show_frame(New
Record ))

bt2=tk.Button(self,text="Manage your
attendance",font=("Times",16),height=2,width=17,command=lambda:controller.show_frame(
Manag eAttendance))

bt3=tk.Button(self,text="Delete your
record",font=("Times",16),height=2,width=17,command=lambda:controller.show_frame(Delet
eReco rd))

bt4=tk.Button(self,text="Edit the
record",font=("Times",16),height=2,width=17,command=lambda:controller.show_frame(EditR
ecord) )
```

```
label1.pack() bt1.pack() bt2.pack() bt3.pack() bt4.pack()

class NewRecord(tk.Frame):
def __init__(self,parent,controller):

tk.Frame.__init__(self,parent)
label1=tk.Label(self,text="New Record",font=("Times",24)) label2=tk.Label(self,text="if you want a new record, previous one will be

deleted,continue?",font=("Times",12))

bt2=tk.Button(self,text="YES",font=("Times",16),height=2,width=17,command=lambda:controller.sh ow_frame(AddSubjects))

bt3=tk.Button(self,text="NO",font=("Times",16),height=2,width=17,command=lambda:controller.show_frame(StartPage))

label1.pack() label2.pack() bt2.pack() bt3.pack()

class ManageAttendance(tk.Frame):
def __init__(self,parent,controller):

tk.Frame.__init__(self,parent)

label1=tk.Label(self,text="Manage Attendance",font=("Times",24))

label1.pack()

bt2=tk.Button(self,text="show status",font=("Times",16),height=2,width=17,command=lambda:self.showstatus(controller))

bt3=tk.Button(self,text="Today's data",font=("Times",16),height=2,width=17,command=lambda:controller.show_frame(TodayData))

bt1=tk.Button(self,text="home",font=("Times",16),height=2,width=17,command=lambda:controller.show_frame(StartPage))

bt2.pack() bt3.pack() bt1.pack()

def showstatus(self,controller):

try:

conn=sql.connect("attend") cur=conn.cursor()

text=""

cur.execute('SELECT * FROM attable')
```

```
for w in cur:
if w[2]==0 and w[3]==0:

per="0" else:

per=w[2]/(w[2]+w[3]) per=per*100 per=str(int(per))

text=text+"sub id "+str(w[0])+" "+w[1]+" "+per+"%\n" messagebox.showinfo("status", text)

except:
messagebox.showinfo("alert!", "There is no record")

class DeleteRecord(tk.Frame):
def __init__(self,parent,controller):

tk.Frame.__init__(self,parent)

label1=tk.Label(self,text="Delete Record",font=("Times",24))

label2=tk.Label(self,text="This action will delete the record,continue?",font=("Times",12))

bt2=tk.Button(self,text="YES",font=("Times",16),height=2,width=17,command=lambda:self.d
elrecord(controller))

bt1=tk.Button(self,text="NO",font=("Times",16),height=2,width=17,command=lambda:contr
oller.show_frame(StartPage))

label1.pack() label2.pack()

bt2.pack()

bt1.pack()
def delrecord(self,controller):

conn=sql.connect('attend') cur=conn.cursor()
cur.execute('DROP TABLE IF EXISTS attable')

conn.commit()
conn.close()

messagebox.showinfo("alert!", "records deleted") controller.show_frame(StartPage)

class EditRecord(tk.Frame):
def __init__(self,parent,controller):

tk.Frame.__init__(self,parent) label1=tk.Label(self,text="Edit Record",font=("Times",24))

bt1=tk.Button(self,text="home",font=("Times",16),height=2,width=17,command=lambda:co
ntroller.show_frame(StartPage))
```

```
label1.pack()
lb2=tk.Label(self,text="input the corresponding subject id",font=("Times",10))
txt1=tk.Entry(self)
lb2.pack()
txt1.pack()
lb3=tk.Label(self,text="number of times attended",font=("Times",10)) txt2=tk.Entry(self)

lb4=tk.Label(self,text="number of times bunked",font=("Times",10)) txt3=tk.Entry(self)

lb3.pack() txt2.pack() lb4.pack() txt3.pack()

bt3=tk.Button(self,text="Update",font=("Times",16),height=2,width=17,command=lambda:s
elf.update(txt1.get(),txt2.get(),txt3.get()))

bt2=tk.Button(self,text="showid of
subjects",font=("Times",16),height=2,width=17,command=lambda:self.showid(controller))

bt2.pack() bt3.pack() bt1.pack()

def update(self,i,p,b): i=int(i)

if p=="" or p==" " or p=="\n": p=0

else:
p=int(p)

if b=="" or b==" " or b=="\n": b=0

else:
b=int(b)

try:

conn=sql.connect("attend") cur=conn.cursor()

cur.execute("SELECT * FROM attable WHERE subid=?",(i,)) kk=cur.fetchone()

np=p

nb=b

cur.execute("UPDATE attable SET attended = ? WHERE subid= ?",(np,i))
cur.execute("UPDATE attable SET bunked = ? WHERE subid= ?",(nb,i)) conn.commit()
conn.close()

messagebox.showinfo("alert!", "Updated")

except:
messagebox.showinfo("alert!", "There is no record")

def showid(self,controller): try:
```

```
conn=sql.connect("attend") cur=conn.cursor() cur.execute('SELECT * FROM attable') text=""

for w in cur:
text=text+"sub id "+str(w[0])+" "+w[1]+"\n"

messagebox.showinfo("subject id", text) conn.commit()
conn.close()

except:
messagebox.showinfo("alert!", "There is no record")

class AddSubjects(tk.Frame):

def __init__(self,parent,controller): tk.Frame.__init__(self,parent)

label1=tk.Label(self,text="add subjects' name seperated by commas(,)",font=("Times",12))

txt1=tk.Text(self,font=("Times",16),width=48,height=3)

bt2=tk.Button(self,text="Add
subjects!",font=("Times",16),height=1,width=17,command=lambda:self.addsub(txt1.get("1.0",
tk.END ),controller))

bt1=tk.Button(self,text="home",font=("Times",16),height=2,width=17,command=lambda:co
ntroller.show_frame(StartPage))

label1.pack() txt1.pack()

bt2.pack()

bt1.pack()
def addsub(self,a,controller):

conn=sql.connect('attend') cur=conn.cursor()
cur.execute('DROP TABLE IF EXISTS attable')

a=a[0:len(a)-1] a=a.split(",")

if len(a)==1 and a[0]=="" :
messagebox.showinfo("alert!", "Please enter the subjects")

else:
sid=1

cur.execute('CREATE TABLE attable(subid INTEGER,subject TEXT,attended

INTEGER,bunked INTEGER)')
for sub in a:

cur.execute('INSERT INTO attable (subid,subject,attended,bunked)
```

```
VALUES(?,?,?,?)',(sid,sub,0,0))
sid=sid+1

conn.commit()
conn.close()
messagebox.showinfo("congratulations!", "subjects are added")
controller.show_frame(StartPage)

class TodayData(tk.Frame):
def __init__(self,parent,controller):

tk.Frame.__init__(self,parent)
label1=tk.Label(self,text="Enter data of today",font=("Times",24))

label1.pack()
subjects",font=("Times",16),height=2,width=17,command=lambda:self.showid(controller))

bt1=tk.Button(self,text="home",font=("Times",16),height=2,width=17,command=lambda:co
ntroller.show_frame(StartPage))

lb2=tk.Label(self,text="input the corresponding subject id",font=("Times",10))
txt1=tk.Entry(self)
lb2.pack()
txt1.pack()

lb3=tk.Label(self,text="number of times attended",font=("Times",10)) txt2=tk.Entry(self)
lb4=tk.Label(self,text="number of times bunked",font=("Times",10)) txt3=tk.Entry(self)

bt2=tk.Button(self,text="showid of

lb3.pack()

txt2.pack()

lb4.pack()
txt3.pack() bt3=tk.Button(self,text="add

to
record",font=("Times",16),height=2,width=17,command=lambda:self.addrecord(txt1.get(),txt2
.get(),

txt3.get()))

bt3.pack() bt2.pack() bt1.pack()

def showid(self,controller): try:

conn=sql.connect("attend") cur=conn.cursor() cur.execute('SELECT * FROM attable') text=""

for w in cur:
text=text+"sub id "+str(w[0])+" "+w[1]+"\n"
```

```
messagebox.showinfo("subject id", text) conn.commit()

conn.close() except:

messagebox.showinfo("alert!", "There is no record") def addrecord(self,i,p,b):

i=int(i)

if p=="" or p==" " or p=="\n": p=0

else:
p=int(p)

if b=="" or b==" " or b=="\n": b=0

else:
b=int(b)

try:

conn=sql.connect("attend")
cur=conn.cursor()
cur.execute("SELECT * FROM attable WHERE subid=?",(i,)) kk=cur.fetchone()
np=kk[2]+p

nb=kk[3]+b

cur.execute("UPDATE attable SET attended = ? WHERE subid= ?",(np,i))
cur.execute("UPDATE attable SET bunked = ? WHERE subid= ?",(nb,i)) conn.commit()
conn.close()

messagebox.showinfo("alert!", "Done") except:

messagebox.showinfo("alert!", "There is no record")

def main(): app=AttendanceManager()

app.title("Attendance Manager") app.mainloop()
if __name__=="__main__": main()
```