LOGIN PAGE

from tkinter import \*  
from tkinter import messagebox  
from PIL import ImageTk  
  
def login():  
 if userEntry.get()=='' or passwordEntry.get()=='':  
 messagebox.showerror('Error','Feilds Cannot be empty')  
 elif userEntry.get()=="Rajat Gupta" and passwordEntry.get()=="12345":  
 messagebox.showinfo('Success','Welcome')  
 window.destroy()  
 import next\_page  
 else:  
 messagebox.showerror('Error','Invalid Credentials')  
  
window = Tk()  
window.geometry('1440x720+0+0')  
window.title('Login Window')  
window.resizable(False,False)  
background = ImageTk.PhotoImage(file = '1695702.jpg')  
bglabel = Label(window,image = background)  
bglabel.place(x=0,y=0)  
  
loginFrame=Frame(window,background='green')  
loginFrame.place(x=600,y=250)  
logoImage = PhotoImage(file='student.png')  
logolabel=Label(loginFrame,image=logoImage)  
logolabel.grid(row=0,column=0,columnspan=2,pady=10)  
  
user = PhotoImage(file = 'username.png')  
usernamelabel = Label(loginFrame,image=user,text = "Username",compound=LEFT,font=('times new roman',20,'bold'))  
usernamelabel.grid(row=1,column=0,pady=10,padx=20)  
userEntry=Entry(loginFrame,font=('times new roman',20,'bold'),bd=8)  
userEntry.grid(row=1,column=1,pady=10,padx=20)  
  
password = PhotoImage(file = 'password.png')  
passwordlabel = Label(loginFrame,image=password,text = "Password",compound=LEFT,font=('times new roman',20,'bold'))  
passwordlabel.grid(row=2,column=0,pady=10,padx=20)  
passwordEntry=Entry(loginFrame,font=('times new roman',20,'bold'),bd=8)  
passwordEntry.grid(row=2,column=1,pady=10,padx=20)  
  
loginbutton = Button(loginFrame,text="Login",font=('times new roman',20,'bold'),width=15,bg='cornflowerblue',cursor='hand2',command=login)  
loginbutton.grid(row=3,column=1)  
  
window.mainloop()

NEXT PAGE

from tkinter import \*  
import ttkthemes  
from tkinter import ttk,messagebox,filedialog  
import pymysql  
import pandas  
import numpy  
#Function  
  
  
def export\_data():  
 url=filedialog.asksaveasfilename(defaultextension='.csv')  
 indexing = student\_table.get\_children()  
 newlist=[]  
 for index in indexing:  
 content=student\_table.item(index)  
 datalist=content['values']  
 newlist.append(datalist)  
  
 table = pandas.DataFrame(newlist,columns=['Roll No.','Name','Outlook Id','Mobile No.','Physics','Chemistry','Maths'])  
 table.to\_csv(url,index=False)  
 messagebox.showinfo('Success','Data Saved successfully')  
  
def update\_student():  
 def update\_data():  
 query = 'update student set Name=%s,OutlookId=%s,MobileNo=%s,Physics=%s,Chemistry=%s,Maths=%s where RollNo=%s'  
 mycursor.execute(query,(nameentry.get(),outlookentry.get(),mobileentry.get(),physicsentry.get(),chemistryentry.get(),mathsentry.get(),rollentry.get()))  
 con.commit()  
 messagebox.showinfo('Success',f'Roll No. {rollentry.get()} if modified successfully',parent=update\_window)  
 update\_window.destroy()  
 show\_student()  
  
 update\_window = Toplevel()  
 update\_window.resizable(0, 0)  
 update\_window.title('Update Student')  
 update\_window.grab\_set()  
 rollnolabel = Label(update\_window, text='Roll No.', font=('times new roman', 20, 'bold'))  
 rollnolabel.grid(row=0, column=0, pady=15, padx=30)  
 rollentry = Entry(update\_window, font=('roman', 15, 'bold'))  
 rollentry.grid(row=0, column=1, pady=15, padx=10)  
  
 namelabel = Label(update\_window, text='Name', font=('times new roman', 20, 'bold'))  
 namelabel.grid(row=1, column=0, pady=15, padx=30)  
 nameentry = Entry(update\_window, font=('roman', 15, 'bold'))  
 nameentry.grid(row=1, column=1, pady=15, padx=10)  
  
 outlooklabel = Label(update\_window, text='Outlook Id', font=('times new roman', 20, 'bold'))  
 outlooklabel.grid(row=2, column=0, pady=15, padx=30)  
 outlookentry = Entry(update\_window, font=('roman', 15, 'bold'))  
 outlookentry.grid(row=2, column=1, pady=15, padx=10)  
  
 mobilelabel = Label(update\_window, text='Mobile', font=('times new roman', 20, 'bold'))  
 mobilelabel.grid(row=3, column=0, pady=15, padx=30)  
 mobileentry = Entry(update\_window, font=('roman', 15, 'bold'))  
 mobileentry.grid(row=3, column=1, pady=15, padx=10)  
  
 physicslabel = Label(update\_window, text='Physics', font=('times new roman', 20, 'bold'))  
 physicslabel.grid(row=4, column=0, pady=15, padx=30)  
 physicsentry = Entry(update\_window, font=('roman', 15, 'bold'))  
 physicsentry.grid(row=4, column=1, pady=15, padx=10)  
  
 chemistrylabel = Label(update\_window, text='Chemistry', font=('times new roman', 20, 'bold'))  
 chemistrylabel.grid(row=5, column=0, pady=15, padx=30)  
 chemistryentry = Entry(update\_window, font=('roman', 15, 'bold'))  
 chemistryentry.grid(row=5, column=1, pady=15, padx=10)  
  
 mathslabel = Label(update\_window, text='Maths', font=('times new roman', 20, 'bold'))  
 mathslabel.grid(row=6, column=0, pady=15, padx=30)  
 mathsentry = Entry(update\_window, font=('roman', 15, 'bold'))  
 mathsentry.grid(row=6, column=1, pady=15, padx=10)  
  
 update\_student\_button = ttk.Button(update\_window, text='UPDATE STUDENT',command=update\_data)  
 update\_student\_button.grid(row=7, columnspan=2, pady=15)  
  
 indexing=student\_table.focus()  
 content=student\_table.item(indexing)  
 listdata=content['values']  
 rollentry.insert(0,listdata[0])  
 nameentry.insert(0,listdata[1])  
 outlookentry.insert(0,listdata[2])  
 mobileentry.insert(0,listdata[3])  
 physicsentry.insert(0,listdata[4])  
 chemistryentry.insert(0,listdata[5])  
 mathsentry.insert(0,listdata[6])  
  
def iexit():  
 result=messagebox.askyesno('Confirm','Do you want to exit?')  
 if result:  
 root.destroy()  
 else:  
 pass  
  
def show\_student():  
 query = 'select \* from student'  
 mycursor.execute(query)  
 fetched\_data = mycursor.fetchall()  
 student\_table.delete(\*student\_table.get\_children())  
 for data in fetched\_data:  
 student\_table.insert('',END,values=data)  
  
def delete\_student():  
 indexing=student\_table.focus()  
 content=student\_table.item(indexing)  
 content\_rollno=content['values'][0]  
 query='delete from student where RollNo=%s'  
 mycursor.execute(query,content\_rollno)  
 con.commit()   
 messagebox.showinfo('Deleted',f'Roll No. {content\_rollno} is deleted successfully')  
 query = 'select \* from student'  
 mycursor.execute(query)  
 fetched\_data = mycursor.fetchall()  
 student\_table.delete(\*student\_table.get\_children())  
 for data in fetched\_data:  
 student\_table.insert('',END,values=data)  
  
  
def search\_student():  
 def search\_data():  
 query = 'select \* from student where RollNo=%s or Name=%s or OutlookId=%s or MobileNo=%s or Physics=%s or Chemistry=%s or Maths=%s'  
 mycursor.execute(query,(rollentry.get(),nameentry.get(),outlookentry.get(),mobileentry.get(),physicsentry.get(),chemistryentry.get(),mathsentry.get()))  
 student\_table.delete(\*student\_table.get\_children())  
 fetched\_data=mycursor.fetchall()  
 for data in fetched\_data:  
 student\_table.insert('',END,values=data)  
  
  
  
  
 search\_window = Toplevel()  
 search\_window.resizable(0, 0)  
 search\_window.title('Search Student')  
 search\_window.grab\_set()  
 rollnolabel = Label(search\_window, text='Roll No.', font=('times new roman', 20, 'bold'))  
 rollnolabel.grid(row=0, column=0, pady=15, padx=30)  
 rollentry = Entry(search\_window, font=('roman', 15, 'bold'))  
 rollentry.grid(row=0, column=1, pady=15, padx=10)  
  
 namelabel = Label(search\_window, text='Name', font=('times new roman', 20, 'bold'))  
 namelabel.grid(row=1, column=0, pady=15, padx=30)  
 nameentry = Entry(search\_window, font=('roman', 15, 'bold'))  
 nameentry.grid(row=1, column=1, pady=15, padx=10)  
  
 outlooklabel = Label(search\_window, text='Outlook Id', font=('times new roman', 20, 'bold'))  
 outlooklabel.grid(row=2, column=0, pady=15, padx=30)  
 outlookentry = Entry(search\_window, font=('roman', 15, 'bold'))  
 outlookentry.grid(row=2, column=1, pady=15, padx=10)  
  
 mobilelabel = Label(search\_window, text='Mobile', font=('times new roman', 20, 'bold'))  
 mobilelabel.grid(row=3, column=0, pady=15, padx=30)  
 mobileentry = Entry(search\_window, font=('roman', 15, 'bold'))  
 mobileentry.grid(row=3, column=1, pady=15, padx=10)  
  
 physicslabel = Label(search\_window, text='Physics', font=('times new roman', 20, 'bold'))  
 physicslabel.grid(row=4, column=0, pady=15, padx=30)  
 physicsentry = Entry(search\_window, font=('roman', 15, 'bold'))  
 physicsentry.grid(row=4, column=1, pady=15, padx=10)  
  
 chemistrylabel = Label(search\_window, text='Chemistry', font=('times new roman', 20, 'bold'))  
 chemistrylabel.grid(row=5, column=0, pady=15, padx=30)  
 chemistryentry = Entry(search\_window, font=('roman', 15, 'bold'))  
 chemistryentry.grid(row=5, column=1, pady=15, padx=10)  
  
 mathslabel = Label(search\_window, text='Maths', font=('times new roman', 20, 'bold'))  
 mathslabel.grid(row=6, column=0, pady=15, padx=30)  
 mathsentry = Entry(search\_window, font=('roman', 15, 'bold'))  
 mathsentry.grid(row=6, column=1, pady=15, padx=10)  
  
 search\_student\_button = ttk.Button(search\_window, text='SEARCH STUDENT', command=search\_data)  
 search\_student\_button.grid(row=7, columnspan=2, pady=15)  
  
def add\_student():  
 def add\_data():  
 if rollentry.get()=='' or mobileentry.get()=='' or outlookentry.get()=='' or nameentry.get()=='' or physicsentry.get()=='' or chemistryentry.get()=='' or mathsentry.get()=='':  
 messagebox.showerror('Error','All fields are required',parent=add\_window)  
 else:  
 query='insert into student values(%s,%s,%s,%s,%s,%s,%s)'  
 mycursor.execute(query,(rollentry.get(),nameentry.get(),outlookentry.get(),mobileentry.get(),physicsentry.get(),chemistryentry.get(),mathsentry.get()))  
 con.commit()  
 result=messagebox.askyesno('Confirm','Data Added. Do you want to clean the form?',parent=add\_window)  
 if result:  
 rollentry.delete(0,END)  
 nameentry.delete(0,END)  
 outlookentry.delete(0,END)  
 mobileentry.delete(0,END)  
 physicsentry.delete(0,END)  
 chemistryentry.delete(0,END)  
 mathsentry.delete(0,END)  
 else:  
 pass  
 query = 'select \* from student'  
 mycursor.execute(query)  
 fetched\_data = mycursor.fetchall()  
 student\_table.delete(\*student\_table.get\_children())  
 for data in fetched\_data:  
 data\_list=list(data)  
 student\_table.insert('',END,values=data\_list)  
  
  
  
 add\_window=Toplevel()  
 add\_window.resizable(0,0)  
 add\_window.grab\_set()  
 rollnolabel=Label(add\_window,text='Roll No.',font=('times new roman',20,'bold'))  
 rollnolabel.grid(row=0,column=0,pady=15,padx=30)  
 rollentry=Entry(add\_window,font=('roman',15,'bold'))  
 rollentry.grid(row=0,column=1,pady=15,padx=10)  
  
 namelabel=Label(add\_window,text='Name',font=('times new roman',20,'bold'))  
 namelabel.grid(row=1,column=0,pady=15,padx=30)  
 nameentry=Entry(add\_window,font=('roman',15,'bold'))  
 nameentry.grid(row=1,column=1,pady=15,padx=10)  
  
 outlooklabel=Label(add\_window,text='Outlook Id',font=('times new roman',20,'bold'))  
 outlooklabel.grid(row=2,column=0,pady=15,padx=30)  
 outlookentry=Entry(add\_window,font=('roman',15,'bold'))  
 outlookentry.grid(row=2,column=1,pady=15,padx=10)  
  
 mobilelabel=Label(add\_window,text='Mobile',font=('times new roman',20,'bold'))  
 mobilelabel.grid(row=3,column=0,pady=15,padx=30)  
 mobileentry=Entry(add\_window,font=('roman',15,'bold'))  
 mobileentry.grid(row=3,column=1,pady=15,padx=10)  
  
 physicslabel=Label(add\_window,text='Physics',font=('times new roman',20,'bold'))  
 physicslabel.grid(row=4,column=0,pady=15,padx=30)  
 physicsentry=Entry(add\_window,font=('roman',15,'bold'))  
 physicsentry.grid(row=4,column=1,pady=15,padx=10)  
  
 chemistrylabel=Label(add\_window,text='Chemistry',font=('times new roman',20,'bold'))  
 chemistrylabel.grid(row=5,column=0,pady=15,padx=30)  
 chemistryentry=Entry(add\_window,font=('roman',15,'bold'))  
 chemistryentry.grid(row=5,column=1,pady=15,padx=10)  
  
 mathslabel=Label(add\_window,text='Maths',font=('times new roman',20,'bold'))  
 mathslabel.grid(row=6,column=0,pady=15,padx=30)  
 mathsentry=Entry(add\_window,font=('roman',15,'bold'))  
 mathsentry.grid(row=6,column=1,pady=15,padx=10)  
  
 add\_student\_button = ttk.Button(add\_window,text='ADD STUDENT',command=add\_data)  
 add\_student\_button.grid(row=7,columnspan=2,pady=15)  
  
count=0  
text=''  
def slider():  
 global text,count  
 if count == len(s):  
 count =0  
 text = ''  
 text=text+s[count]  
 sliderLabel.config(text=text)  
 count+=1  
 sliderLabel.after(250,slider)  
  
def connect\_to\_database():  
 def connect():  
 global mycursor,con  
 try:  
 con = pymysql.connect(host=hostentry.get(), user=userentry.get(), password=passwordentry.get())  
 mycursor = con.cursor()  
 messagebox.showinfo('Success','Database Connection Successfull',parent=connectwindow)  
 except:  
 messagebox.showerror('Error','Invalid Details',parent=connectwindow)  
 try:  
 query ='Create database studentinformationsystem'  
 mycursor.execute(query)  
 query='use studentinformationsystem'  
 mycursor.execute(query)  
 query='create table student(RollNo int not null primary key,Name varchar(30),OutlookId varchar(30),MobileNo varchar(10),Physics varchar(3),Chemistry varchar(3),Maths varchar(3))'  
 mycursor.execute(query)  
 except:  
 query='use studentinformationsystem'  
 mycursor.execute(query)  
 messagebox.showinfo('Success','Database Connection is successful',parent = connectwindow)  
 connectwindow.destroy()  
 addstudentbutton.config(state=NORMAL)  
 searchstudentbutton.config(state=NORMAL)  
 updatestudentbutton.config(state=NORMAL)  
 showstudentbutton.config(state=NORMAL)  
 exportstudentbutton.config(state=NORMAL)  
 deletestudentbutton.config(state=NORMAL)  
  
 connectwindow = Toplevel()  
 connectwindow.grab\_set()  
 connectwindow.geometry('470x250+730+230')  
 connectwindow.title('Database Connection')  
 connectwindow.resizable(0,0)  
  
 hostnamelabel = Label(connectwindow,text="Host Name",font=('arial',20,'bold'))  
 hostnamelabel.grid(row=0,column=0,padx=20)  
  
 hostentry = Entry(connectwindow,font=('roman',15,'bold'),bd=2)  
 hostentry.grid(row=0,column=1,padx=40,pady=20)  
  
 usernamelabel = Label(connectwindow,text="User Name",font=('arial',20,'bold'))  
 usernamelabel.grid(row=1,column=0,padx=20)  
  
 userentry = Entry(connectwindow,font=('roman',15,'bold'),bd=2)  
 userentry.grid(row=1,column=1,padx=40,pady=20)  
  
 passwordnamelabel = Label(connectwindow,text="Password",font=('arial',20,'bold'))  
 passwordnamelabel.grid(row=2,column=0,padx=20)  
  
 passwordentry = Entry(connectwindow,font=('roman',15,'bold'),bd=2)  
 passwordentry.grid(row=2,column=1,padx=40,pady=20)  
  
 connectbutton = ttk.Button(connectwindow,text="CONNECT",command=connect)  
 connectbutton.grid(row=3,columnspan=2)  
  
  
  
#GUI  
  
root = ttkthemes.ThemedTk()  
root.get\_themes()  
root.set\_theme('radiance')  
root.geometry('1174x680+0+0')  
root.title('Student Information System')  
root.resizable(0,0)  
  
s = 'Student Information System'  
sliderLabel = Label(root,text =s,font=('arial',28,'italic bold'),width=30)  
sliderLabel.place(x=200,y=0)  
slider()  
  
connectbutton = ttk.Button(root,text='Connect To DataBase',command=connect\_to\_database)  
connectbutton.place(x=990,y=0)  
  
leftframe = Frame(root)  
leftframe.place(x=50,y=80,width=300,height=600)  
  
logo\_image = PhotoImage(file='studentlogo.png')  
logo\_label = Label(leftframe,image=logo\_image)  
logo\_label.grid(row=0,column=0)  
  
addstudentbutton = ttk.Button(leftframe,text='Add Student',width=25,state=DISABLED,command=add\_student)  
addstudentbutton.grid(row=1,column=0,pady=20)  
  
searchstudentbutton = ttk.Button(leftframe,text='Search Student',width=25,state=DISABLED,command=search\_student)  
searchstudentbutton.grid(row=2,column=0,pady=20)  
  
deletestudentbutton = ttk.Button(leftframe,text='Delete Student',width=25,state=DISABLED,command=delete\_student)  
deletestudentbutton.grid(row=3,column=0,pady=20)  
  
updatestudentbutton = ttk.Button(leftframe,text='Update Student',width=25,state=DISABLED,command=update\_student)  
updatestudentbutton.grid(row=4,column=0,pady=20)  
  
showstudentbutton = ttk.Button(leftframe,text='Show Student',width=25,state=DISABLED,command=show\_student)  
showstudentbutton.grid(row=5,column=0,pady=20)  
  
exportstudentbutton = ttk.Button(leftframe,text='Export Student',width=25,state=DISABLED,command=export\_data)  
exportstudentbutton.grid(row=6,column=0,pady=20)  
  
exitbutton = ttk.Button(leftframe,text='Exit',width=25,command=iexit)  
exitbutton.grid(row=7,column=0,pady=20)  
  
rightframe = Frame(root)  
rightframe.place(x=350,y=80,width=820,height=600)  
  
scroll\_x=Scrollbar(rightframe,orient=HORIZONTAL)  
scroll\_y=Scrollbar(rightframe,orient=VERTICAL)  
  
student\_table=ttk.Treeview(rightframe,columns=('RollNo','Name','OutlookId','MobileNo','Physics','Chemistry','Maths'),xscrollcommand=scroll\_x.set,yscrollcommand=scroll\_y.set)  
  
scroll\_x.config(command=student\_table.xview)  
scroll\_y.config(command=student\_table.yview)  
  
scroll\_x.pack(side=BOTTOM,fill=X)  
scroll\_y.pack(side=RIGHT,fill=Y)  
  
student\_table.pack(fill=BOTH,expand=1)  
  
student\_table.heading('RollNo',text='Roll No.')  
student\_table.heading('Name',text='Name')  
student\_table.heading('OutlookId',text='Outlook Id')  
student\_table.heading('MobileNo',text='Mobile No.')  
student\_table.heading('Physics',text='Physics')  
student\_table.heading('Chemistry',text='Chemistry')  
student\_table.heading('Maths',text='Maths')  
  
student\_table.config(show='headings')  
  
root.mainloop()