from tkinter import \*  
from tkinter import ttk  
import pymysql  
from tkinter import messagebox  
  
class Student:  
 def \_\_init\_\_(self , root):  
 self.root = root  
 self.root.title(" Student Management System ")  
 self.root.geometry('1350x700')  
  
  
 title = Label(self.root,text=" Student Management System " ,bd=20, relief= GROOVE ,font = ("times new roman",35,"bold"), bg="#50C878", fg="#3F704D" )  
 title.pack(side=TOP , fill=X)  
  
 #============All variables==============  
 self.Roll\_No\_var=StringVar()  
 self.name\_var=StringVar()  
 self.email\_var=StringVar()  
 self.gender\_var=StringVar()  
 self.contact\_var=StringVar()  
 self.dob\_var=StringVar()  
  
  
  
 self.search\_by = StringVar()  
 self.search\_txt = StringVar()  
  
  
  
 #=====================Manage Frame==========================#  
  
 Manage\_Frame=Frame(self.root, bd=4, relief=RIDGE, bg='#FF7417')  
 Manage\_Frame.place(x=10,y=100,width=480,height=600)  
  
 m\_title=Label(Manage\_Frame,text="Manage Students", bg='#FF7417', fg='#6F2DA8', font=("times new roman",28,"bold"))  
 m\_title.grid(row=0,columnspan=2,pady=30)  
  
 lb1\_roll=Label(Manage\_Frame,text="Roll No", bg='#FF7417', fg='#FFF200', font=("times new roman",22,"bold"))  
 lb1\_roll.grid(row=1,column=0,pady=10,padx=20, sticky="w")  
  
 txt\_Roll=Entry(Manage\_Frame, textvariable=self.Roll\_No\_var,font=("times new roman",15,"bold"),bd=5,relief=GROOVE)  
 txt\_Roll.grid(row=1,column=1,pady=10,padx=20, sticky="w")  
  
 lb1\_name = Label(Manage\_Frame, text="Name", bg='#FF7417', fg='#FFF200', font=("times new roman", 22, "bold"))  
 lb1\_name.grid(row=2, column=0, pady=10, padx=20, sticky="w")  
  
 txt\_name = Entry(Manage\_Frame, textvariable=self.name\_var,font=("times new roman", 15, "bold"), bd=5, relief=GROOVE)  
 txt\_name.grid(row=2, column=1, pady=10, padx=20, sticky="w")  
  
 lb1\_email = Label(Manage\_Frame, text="Email", bg='#FF7417', fg='#FFF200', font=("times new roman", 22, "bold"))  
 lb1\_email.grid(row=3, column=0, pady=10, padx=20, sticky="w")  
  
 txt\_email= Entry(Manage\_Frame, textvariable=self.email\_var,font=("times new roman", 15, "bold"), bd=5, relief=GROOVE)  
 txt\_email.grid(row=3, column=1, pady=10, padx=20, sticky="w")  
  
 lb1\_Gender = Label(Manage\_Frame, text="Gender", bg='#FF7417', fg='#FFF200', font=("times new roman", 22, "bold"))  
 lb1\_Gender.grid(row=4, column=0, pady=10, padx=20, sticky="w")  
  
 combo\_gender=ttk.Combobox(Manage\_Frame,textvariable=self.gender\_var,font=("times new roman", 13, "bold"),state='readonly')  
 combo\_gender['values']=("Male","Female","Other")  
 combo\_gender.grid(row=4, column=1, padx=20, pady=10)  
  
  
 lb1\_dob= Label(Manage\_Frame, text="DOB", bg='#FF7417', fg='#FFF200', font=("times new roman",22, "bold"))  
 lb1\_dob.grid(row=5, column=0, pady=10, padx=20, sticky="w")  
  
 txt\_dob = Entry(Manage\_Frame, textvariable=self.dob\_var,font=("times new roman", 15, "bold"), bd=5, relief=GROOVE)  
 txt\_dob.grid(row=5, column=1, pady=10, padx=20, sticky="w")  
  
 lb1\_contact = Label(Manage\_Frame, text="Contact", bg='#FF7417', fg='#FFF200', font=("times new roman", 22, "bold"))  
 lb1\_contact.grid(row=6, column=0, pady=10, padx=20, sticky="w")  
  
 txt\_contact = Entry(Manage\_Frame,textvariable=self.contact\_var,font=("times new roman", 15, "bold"), bd=5, relief=GROOVE)  
 txt\_contact.grid(row=6, column=1, pady=10, padx=20, sticky="w")  
  
 lb1\_address = Label(Manage\_Frame, text="Address ", bg='#FF7417', fg='#FFF200', font=("times new roman", 22, "bold"))  
 lb1\_address.grid(row=7, column=0, pady=10, padx=20, sticky="w")  
  
 self.txt\_address = Text(Manage\_Frame, width=30 , height= 3, font=("", 10))  
 self.txt\_address.grid(row=7, column=1, pady=10, padx=20, sticky="w")  
  
 # =======Button Frame=======  
  
 btn\_Frame=Frame(Manage\_Frame,bd=5, relief=RIDGE,bg='red')  
 btn\_Frame.place(x=15 ,y=535, width=450)  
  
 Addbtn=Button(btn\_Frame,text='Add',width=10,command=self.add\_students).grid(row=0,column=0,padx=10,pady=10)  
 updatebtn = Button(btn\_Frame, text='Update', width=10,command=self.update\_data).grid(row=0, column=1, padx=10, pady=10)  
 deletebtn = Button(btn\_Frame, text='Delete', width=10,command=self.delete\_data).grid(row=0, column=2, padx=10, pady=10)  
 clearbtn = Button(btn\_Frame, text='Clear', width=10,command=self.clear).grid(row=0, column=3, padx=10, pady=10)  
  
  
  
  
  
  
  
  
  
 # =====================Detail Frame==========================#  
  
 Detail\_Frame = Frame(self.root, bd=4, relief=RIDGE, bg='sky blue')  
 Detail\_Frame.place(x=500, y=100, width=840, height=600)  
  
  
  
 lb1\_search=Label(Detail\_Frame,text='Search By',bg='sky blue', fg='white',font=("times new roman",20,"bold"))  
 lb1\_search.grid(row=0,column=0,padx=10,pady=20,sticky='w')  
  
 combo\_search = ttk.Combobox(Detail\_Frame, textvariable=self.search\_by,font=("times new roman", 13, "bold"), state='readonly')  
 combo\_search['values'] = ("Roll\_No", "Name", "Contact")  
 combo\_search.grid(row=0, column=1, padx=20, pady=10)  
  
 txt\_search = Entry(Detail\_Frame,textvariable=self.search\_txt, width=20 , font=("times new roman", 13, "bold"), bd=5, relief=GROOVE)  
 txt\_search.grid(row=0, column=2, pady=10, padx=20, sticky="w")  
  
 searchbtn = Button(Detail\_Frame, text='Search', width=10 ,pady=5,command=self.search\_by).grid(row=0, column=3, padx=10, pady=10)  
 showallbtn = Button(Detail\_Frame, text='Show All', width=10 ,pady=5,command=self.search\_txt).grid(row=0, column=4, padx=10, pady=10)  
  
  
 #=============Table Frame=============  
  
 Table\_Frame=Frame(Detail\_Frame,bd=4,relief=RIDGE,bg='sky blue')  
 Table\_Frame.place(x=10,y=70,width=810 , height=500)  
  
 scroll\_x=Scrollbar(Table\_Frame,orient=HORIZONTAL)  
 scroll\_y= Scrollbar(Table\_Frame, orient=VERTICAL)  
  
 self.Student\_table=ttk.Treeview(Table\_Frame,columns=("roll","name","email", "gender" , "contact" , "dob", "address"), xscrollcommand=scroll\_x.set, yscrollcommand=scroll\_y.set)  
 scroll\_x.pack(side=BOTTOM, fill=X)  
 scroll\_y.pack(side=RIGHT, fil=Y)  
 scroll\_x.config(command=self.Student\_table.xview)  
 scroll\_y.config(command=self.Student\_table.yview)  
  
 self.Student\_table.heading("roll",text='Roll no')  
 self.Student\_table.heading("name",text='Name')  
 self.Student\_table.heading("email",text='Email')  
 self.Student\_table.heading("gender",text='Gender')  
 self.Student\_table.heading("contact",text='Contact')  
 self.Student\_table.heading("dob",text='D.O.B')  
 self.Student\_table.heading("address",text='Address')  
 self.Student\_table['show'] = 'headings'  
 self.Student\_table.column("roll",width=50)  
 self.Student\_table.column("name", width=250)  
 self.Student\_table.column("email", width=350)  
 self.Student\_table.column("gender", width=150)  
 self.Student\_table.column("contact", width=250)  
 self.Student\_table.column("dob", width=200)  
 self.Student\_table.column("address", width=500)  
 self.Student\_table.pack(fill=BOTH, expand=1)  
 self.Student\_table.bind("<ButtonRelease-1>", self.get\_cursor)  
 self.fetch\_data()  
  
  
 def add\_students(self):  
 if self.Roll\_No\_var.get()=="" or self.name\_var.get()=="" :  
 messagebox.showerror("Error","All fields are required !!!!")  
 else:  
 #  
 con=pymysql.connect(host="localhost", user="root",password="",database="stm1")  
 cur=con.cursor()  
 cur.execute("insert into students values(%s,%s,%s,%s,%s,%s,%s)",(self.Roll\_No\_var.get(),  
 self.name\_var.get(),  
 self.email\_var.get(),  
 self.gender\_var.get(),  
 self.contact\_var.get(),  
 self.dob\_var.get(),  
 self.txt\_address.get('1.0',END)  
 ))  
 con.commit()  
 self.fetch\_data()  
 self.clear()  
 con.close()  
 messagebox.showinfo("Sucess", "Record has been insreted")  
  
 def fetch\_data(self):  
 con = pymysql.connect(host="localhost", user="root", password="", database="stm1")  
 cur = con.cursor()  
 cur.execute("select \* from students")  
 rows=cur.fetchall()  
 if len(rows)!=0:  
 self.Student\_table.delete(\*self.Student\_table.get\_children())  
 for row in rows:  
 self.Student\_table.insert('',END,values=row)  
 con.commit()  
 con.close()  
  
 def clear(self):  
 self.Roll\_No\_var.set("")  
 self.name\_var.set("")  
 self.email\_var.set("")  
 self.gender\_var.set("")  
 self.contact\_var.set("")  
 self.dob\_var.set("")  
 self.txt\_address.delete("1.0",END)  
  
 def get\_cursor(self,ev):  
 cursor\_row=self.Student\_table.focus()  
 contents=self.Student\_table.item(cursor\_row)  
 row=contents['values']  
 self.Roll\_No\_var.set(row[0])  
 self.name\_var.set(row[1])  
 self.email\_var.set(row[2])  
 self.gender\_var.set(row[3])  
 self.contact\_var.set(row[4])  
 self.dob\_var.set(row[5])  
 self.txt\_address.delete("1.0", END)  
 self.txt\_address.insert(END,row[6])  
  
 def update\_data(self):  
 con = pymysql.connect(host="localhost", user="root", password="", database="stm1")  
 cur = con.cursor()  
 cur.execute("update students set name=%s,email=%s,gender=%s,contact=%s,dob=%s,address=%s, where roll\_no= %s",(  
 self.name\_var.get(),  
 self.email\_var.get(),  
 self.gender\_var.get(),  
 self.contact\_var.get(),  
 self.dob\_var.get(),  
 self.txt\_address.get('1.0', END),  
 self.Roll\_No\_var.get()  
 ))  
 con.commit()  
 self.fetch\_data()  
 self.clear()  
 con.close()  
  
  
 def delete\_data(self):  
 con = pymysql.connect(host="localhost", user="root", password="", database="stm1")  
 cur = con.cursor()  
 cur.execute("delete from students where roll\_no=%s" , self.Roll\_No\_var.get())  
 con.commit()  
 con.close()  
 self.fetch\_data()  
 self.clear()  
  
 def search\_data(self):  
 con = pymysql.connect(host="localhost", user="root", password="", database="stm1")  
 cur = con.cursor()  
 cur.execute("select \* from students where " + str(self.search\_by.get())+ "LIKE '%" + str(self.search\_txt.get()) + " %' ")  
 rows=cur.fetchall()  
 if len(rows)!=0:  
 self.Student\_table.delete(\*self.Student\_table.get\_children())  
 for row in rows:  
 self.Student\_table.insert('',END,values=row)  
 con.commit()  
 con.close()  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
root = Tk()  
  
ob=Student(root)  
  
root.mainloop()