SOURCE CODE

```
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")
        root.geometry("1350x700+0+0")

        title = Label(self.root, text="Student Management System", bd=10, relief=GROOVE,
font=("ENTER FONT NAME HERE", 40, "bold"), bg=" ENTER THE NAME OF COLOR ", fg=" ENTER THE
NAME OF COLOR

")
        title.pack(side=TOP, fill=X)


        self.roll_var = StringVar()
        self.name_var = StringVar()
        self.email_var = StringVar()
        self.gender_var = StringVar()
        self.contact_var = StringVar()
        self.dob_var = StringVar()
        self.txt_Address = StringVar()

        self.search_by = StringVar()
        self.search_txt = StringVar()



        Manage_Frame = Frame(self.root, bd=4, relief=RIDGE, bg="gray")
        Manage_Frame.place(x=20, y=100, width=450, height=600)

        m_title = Label(Manage_Frame, text="Manage Student",  bg=" ENTER THE NAME OF COLOR
", fg=" ENTER THE NAME OF COLOR ", font=("ENTER FONT NAME HERE", 20, "bold"))
        m_title.grid(row=0, columnspan=2, pady=20)

        lbl_roll = Label(Manage_Frame, text="Roll No.",  bg=" ENTER THE NAME OF COLOR ",
fg=" ENTER THE NAME OF COLOR ", font=(" ENTER FONT NAME HERE ", 18, "bold"))
        lbl_roll.grid(row=1, column=0, padx=20, pady=10, sticky="w")
        txt_roll = Entry(Manage_Frame, textvariable=self.roll_var, font=(" ENTER FONT NAME
HERE ", 18, "bold"), bd=5, relief=GROOVE)
        txt_roll.grid(row=1, column=1, padx=20, pady=10, sticky="w")

        lbl_name = Label(Manage_Frame, text="Name",  bg=" ENTER THE NAME OF COLOR ", fg="
ENTER THE NAME OF COLOR ", font=(" ENTER FONT NAME HERE ", 18, "bold"))
        lbl_name.grid(row=2, column=0, padx=20, pady=10, sticky="w")
        txt_name = Entry(Manage_Frame, textvariable=self.name_var, font=(" ENTER FONT NAME
HERE ", 18, "bold"), bd=5, relief=GROOVE)
```

```
        txt_name.grid(row=2, column=1, padx=20, pady=10, sticky="w")

        lbl_email = Label(Manage_Frame, text="Email",  bg=" ENTER THE NAME OF COLOR ", fg=" ENTER THE NAME OF COLOR ", font=(" ENTER FONT NAME HERE ", 18, "bold"))
        lbl_email.grid(row=3, column=0, padx=20, pady=10, sticky="w")
        txt_email = Entry(Manage_Frame, textvariable=self.email_var,  font=(" ENTER FONT NAME HERE ", 18, "bold"), bd=5, relief=GROOVE)
        txt_email.grid(row=3, column=1, padx=20, pady=10, sticky="w")

        lbl_gender = Label(Manage_Frame, text="Gender",  bg="gray", fg=" ENTER THE NAME OF COLOR ", font=(" ENTER FONT NAME HERE ", 18, "bold"))
        lbl_gender.grid(row=4, column=0, padx=20, pady=10, sticky="w")
        combo_gender = ttk.Combobox(Manage_Frame, textvariable=self.gender_var, font=(" ENTER FONT NAME HERE ", 13, "bold"), state="readonly" )
        combo_gender['values'] = ("male", "female", "other")
        combo_gender.grid(row=4, column=1, padx=20, pady=10, sticky="w")

        lbl_contact = Label(Manage_Frame, text="Contact",  bg="gray", fg=" ENTER THE NAME OF COLOR ", font=(" ENTER FONT NAME HERE ", 18, "bold"))
        lbl_contact.grid(row=5, column=0, padx=20, pady=10, sticky="w")
        txt_contact = Entry(Manage_Frame, textvariable=self.contact_var, font=(" ENTER FONT NAME HERE ", 18, "bold"), bd=5, relief=GROOVE)
        txt_contact.grid(row=5, column=1, padx=20, pady=10, sticky="w")


        lbl_dob = Label(Manage_Frame, text="DOB",  bg=" ENTER THE NAME OF COLOR ", fg=" ENTER THE NAME OF COLOR ", font=(" ENTER FONT NAME HERE ", 18, "bold"))
        lbl_dob.grid(row=6, column=0, padx=20, pady=10, sticky="w")
        txt_gender = Entry(Manage_Frame, textvariable=self.dob_var, font=(" ENTER FONT NAME HERE ", 18, "bold"), bd=5, relief=GROOVE)
        txt_gender.grid(row=6, column=1, padx=20, pady=10, sticky="w")

        lbl_address = Label(Manage_Frame, text="Address",  bg=" ENTER THE NAME OF COLOR ", fg=" ENTER THE NAME OF COLOR ", font=(" ENTER FONT NAME HERE ", 18, "bold"))
        lbl_address.grid(row=7, column=0, padx=20, pady=10, sticky="w")
        self.txt_Address =Text(Manage_Frame, width=30, height=4, font=("", 10))
        self.txt_Address.grid(row=7, column=1, padx=20, pady=10, sticky="w")

        #button frame
        Button_Frame = Frame(Manage_Frame, bd=4, relief=RIDGE, bg=" ENTER THE NAME OF COLOR ")
        Button_Frame.place(x=10, y=510, width=430)

        addbtn = Button(Button_Frame, text="Add", width=8, command=self.add_student).grid(row=0, column=0, padx=20, pady=5)
        updatebtn = Button(Button_Frame, text="Update", width=8, command=self.update_data).grid(row=0, column=1, padx=20, pady=5)
        deletebtn = Button(Button_Frame, text="Delete", width=8, command=self.delete_data).grid(row=0, column=2, padx=20, pady=5)
        clearbtn = Button(Button_Frame, text="Clear", width=8, command=self.clear).grid(row=0, column=3, padx=20, pady=5)


        #==============Detail Frame=======================
        Detail_Frame = Frame(self.root, bd=4, relief=RIDGE, bg=" ENTER THE NAME OF COLOR
")
        Detail_Frame.place(x=500, y=100, width=750, height=580)

        lbl_search = Label(Detail_Frame, width=8, text="Search By",  bg=" ENTER THE NAME
OF COLOR ", fg=" ENTER THE NAME OF COLOR ", font=("ENTER FONT NAME", 15, "bold"))
        lbl_search.grid(row=0, column=0, padx=5, pady=5, sticky="w")
        combo_search = ttk.Combobox(Detail_Frame, textvariable=self.search_by,
font=("ENTER FONT NAME", 13), state="readonly" )
        combo_search['values'] = ("Roll_no")
        combo_search.grid(row=0, column=1, padx=5, pady=5, sticky="w")

        txt_search = Entry(Detail_Frame, textvariable=self.search_txt, font=(" ENTER FONT
NAME HERE ", 13), bd=5, relief=GROOVE)
        txt_search.grid(row=0, column=2, padx=5, pady=5, sticky="w")

        searchbtn = Button(Detail_Frame, text="Search", width=8,
command=self.search_data).grid(row=0, column=3, padx=5, pady=5)
        showallbtn = Button(Detail_Frame, text="Show All",
width=8,  command=self.fetch_data).grid(row=0, column=4, padx=5, pady=5)

        #Table Frame
        Table_Frame = Frame(Detail_Frame, bd=4, relief=RIDGE, bg="ENTER THE NAME OF
COLOR")
        Table_Frame.place(x=20, y=60, width=700, 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, fill=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="DOB :")
        self.Student_table.heading("address", text="Address")
        self.Student_table['show']='headings' # removing extra index col at begining

        #setting up widths of cols
        self.Student_table.column("roll", width=100)
        self.Student_table.column("name", width=100)
        self.Student_table.column("email", width=100)
        self.Student_table.column("gender", width=100)
        self.Student_table.column("contact", width=100)
        self.Student_table.column("dob", width=100)
        self.Student_table.column("address", width=100)
        self.Student_table.pack(fill=BOTH, expand=1) #fill both is used to fill cols
around the frame
        self.Student_table.bind("<ButtonRelease-1>", self.get_cursor)# this is an event to
select row

        self.fetch_data() #to display data in grid

    def add_student(self):

        if self.roll_var.get()=="" or self.name_var.get()=="":
            messagebox.showerror("Error", "please fill all the fields!!!")

        else:
            con = pymysql.connect(host="localhost", user="root", password="root",
database="ENTER NAME OF DATABASE")
            cur = con.cursor()

            cur.execute("insert into info values(%s, %s, %s, %s, %s, %s, %s)",
(self.roll_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("Successfull", "Record has been inserted.")

    def fetch_data(self):

        con = pymysql.connect(host="localhost", user="root", password="root",
database="ENTER NAME OF DATABASE")
        cur = con.cursor()

        cur.execute("select * from info")
        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_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, evnt):
        cursor_row = self.Student_table.focus()
        content = self.Student_table.item(cursor_row)
        row = content['values']
        self.roll_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])
#error here
    def update_data(self):
        if self.roll_var.get()=="" or self.name_var.get()=="":
            messagebox.showerror("Error", "please fill all the fields!!!")
        else:
            con = pymysql.connect(host="localhost", user="root", password="root",
database="ENTER NAME OF YOUR DATABASE")
            cur = con.cursor()

            cur.execute("UPDATE info SET name=%s, email=%s, gender=%s, contact=%s, dob=%s,
address=%s where roll=%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_var.get(),))


            con.commit()
            self.fetch_data() #
            self.clear()
            con.close()
            messagebox.showinfo("successfull", "Record has been updated.")

    def delete_data(self):
        con = pymysql.connect(host="localhost", user="root", password="root",
database="ENTER NAME OF YOUR DATABASE")
        cur = con.cursor()

        cur.execute("delete from info where roll=%s",self.roll_var.get())
        con.commit()
        self.fetch_data()
        self.clear()
        con.close()
        messagebox.showinfo("successfull", "Record has been deleted.")




    def search_data(self):

        con = pymysql.connect(host="localhost", user="root", password="root",
database="ENTER NAME OF YOUR DATABASE")
        cur = con.cursor()
        sql = "SELECT * FROM info WHERE roll = %s"
        adr = self.search_txt.get()

        val = cur.execute(sql, adr)
        if(not val):
            messagebox.showinfo("No", "Not availabe!")



        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()
obj = Student(root)
root.mainloop()
```

DATABASE QUERIES:

1.create database <database_name>;

2.use <database_name>;

3.create table <table_name>(contain of table with datatype(use varchar datatype));

4. desc <table_name>; 🡐-------and check the table properly

5.To insert:

Insert <table_name> values('type the info which you want to store in the table');