UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

CU CHANDIGARH UNIVERSITY

NAAC GRADE A+ ACCREDITED UNIVERSITY

Student Name: Diksha
UID: 24MCI10238
Branch: MCA (AIML)
Section/Group: 24MAM-4A
Semester: 1st
Date of Performance: 23/10/2024
Subject Name: Python Programming
Subject Code: 24CAH-606

Q. Contact Book.

Aim/Overview of the project:

The project aims to create a simple contact book application using Python's Tkinter library. The application allows users to add, view, update, and delete contacts, providing an intuitive graphical user interface (GUI) for managing contact information.

1. Task to be done:

- **Set Up GUI :**

  Initialize the main application window.
  Design the layout including labels, entry fields, buttons, and a listbox to display contacts.

- **Implement Core Functions**:

  Create functions to add new contacts, view selected contacts, update existing contacts, and delete contacts from the list.
- Implement data validation to ensure that users enter valid information before performing actions.

- **Manage Contact List**:

- Store contacts in a list format, allowing for easy manipulation (adding, updating, deleting). Sort and display the contact list in alphabetical order.

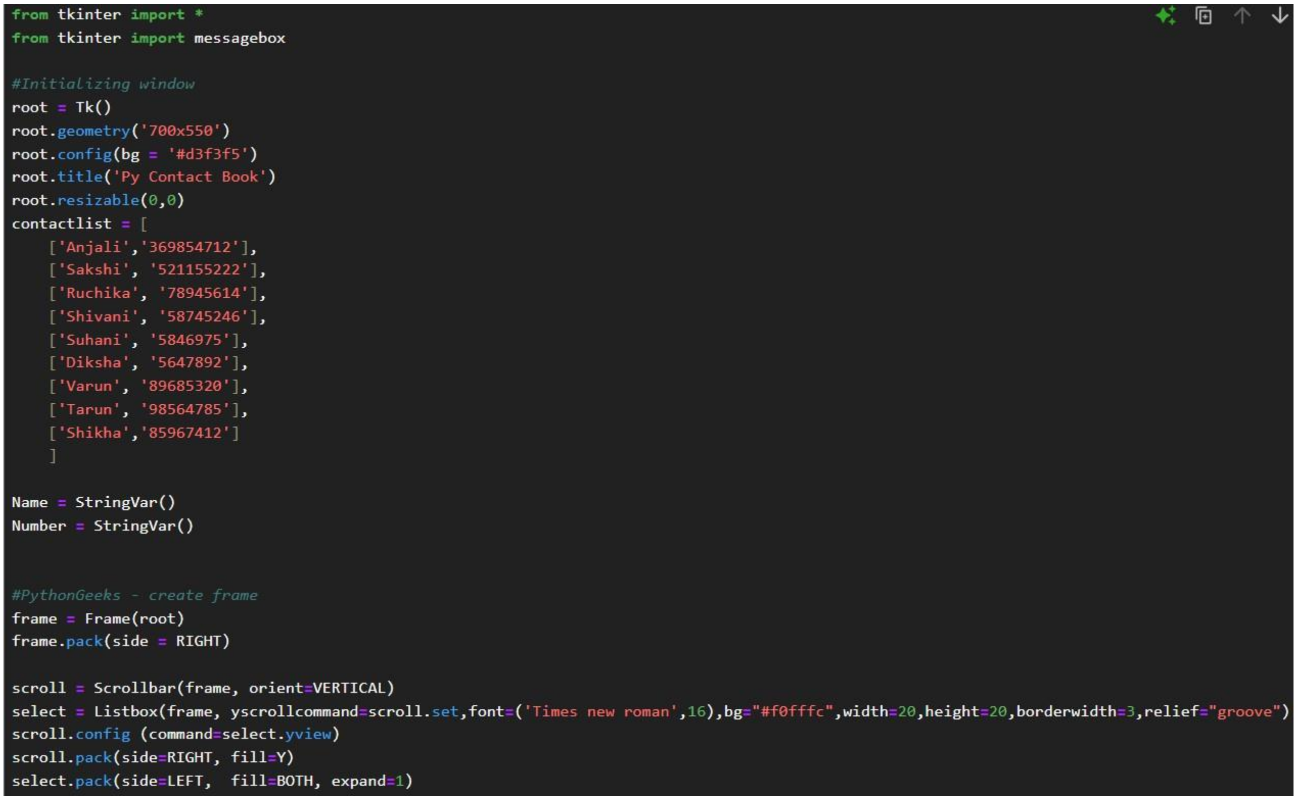
- **Integrate User Feedback**:

- Use message boxes to provide feedback for actions (confirmation messages for successful operations and error messages for invalid actions).

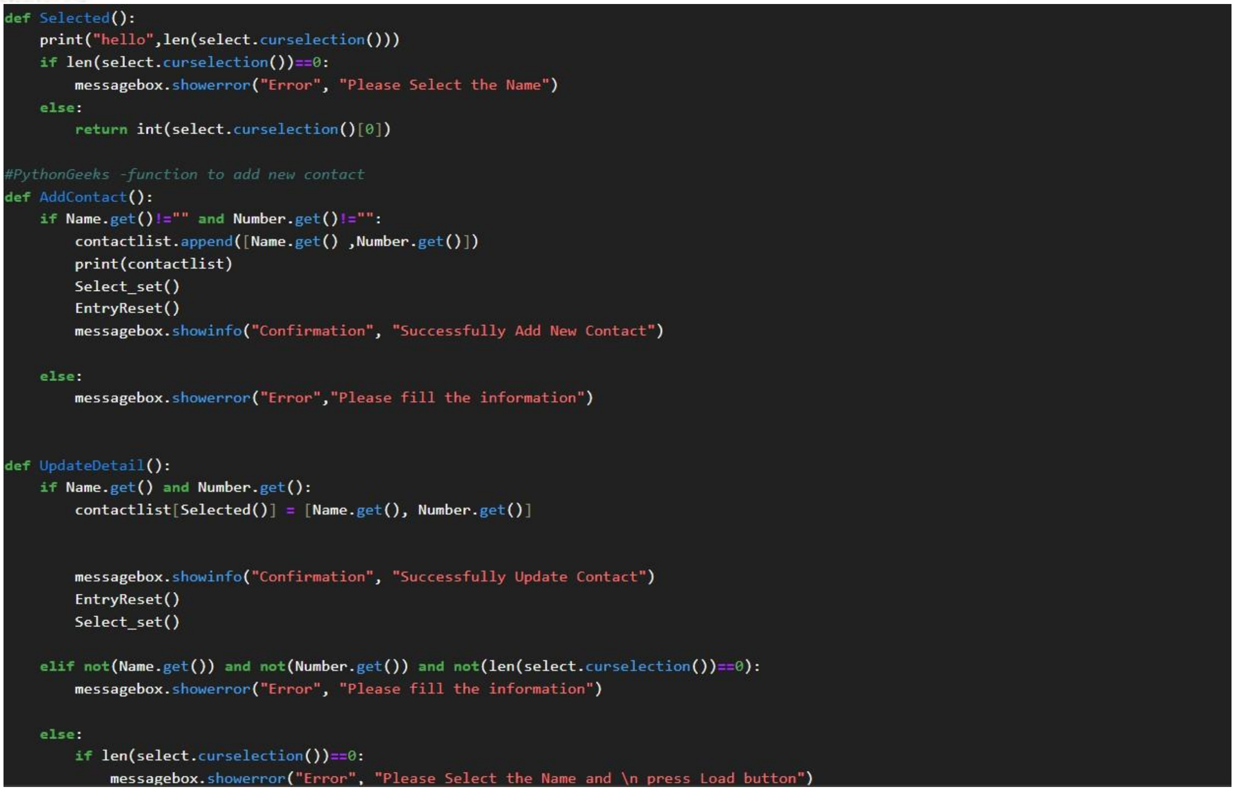
Testing and Debugging:

Ensure that all functionalities work correctly and handle edge cases (e.g., selecting a contact, empty input fields).

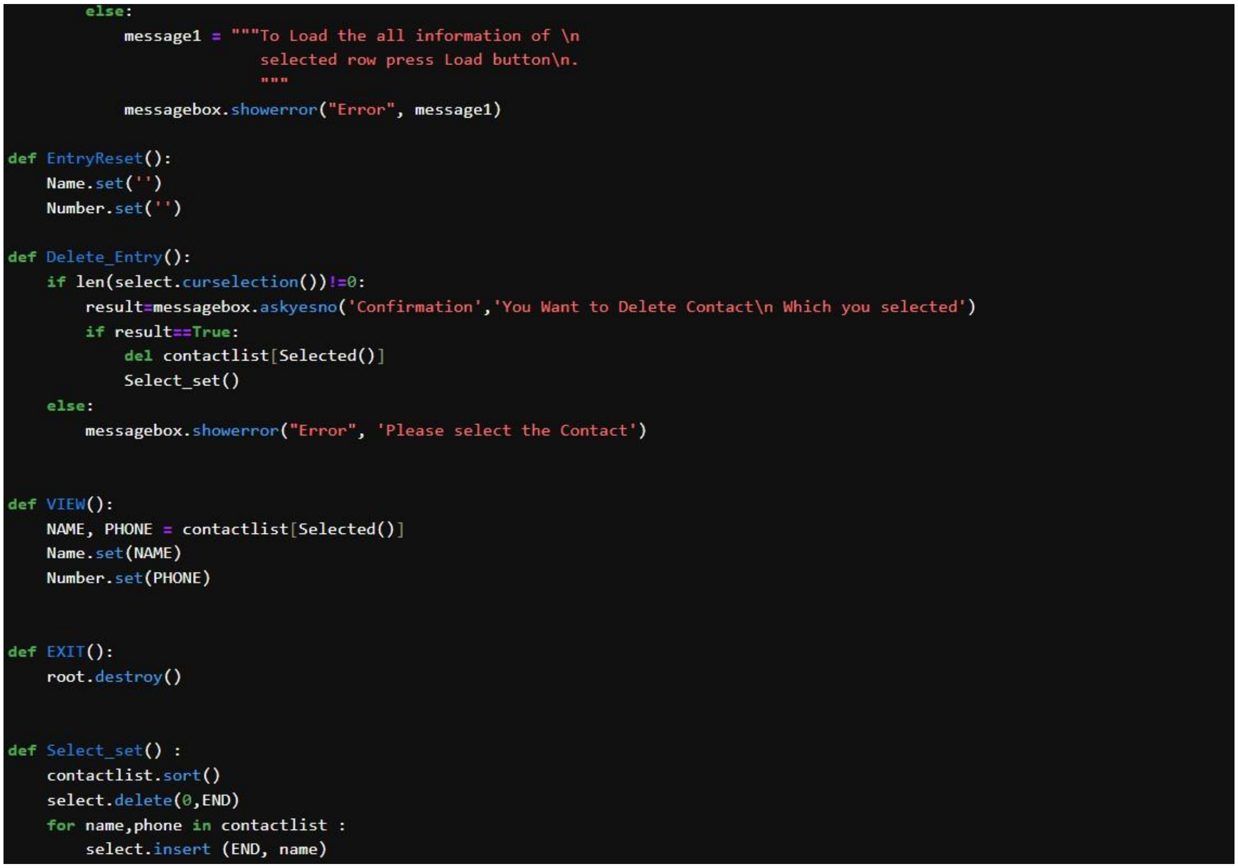
2. Code For Project:-



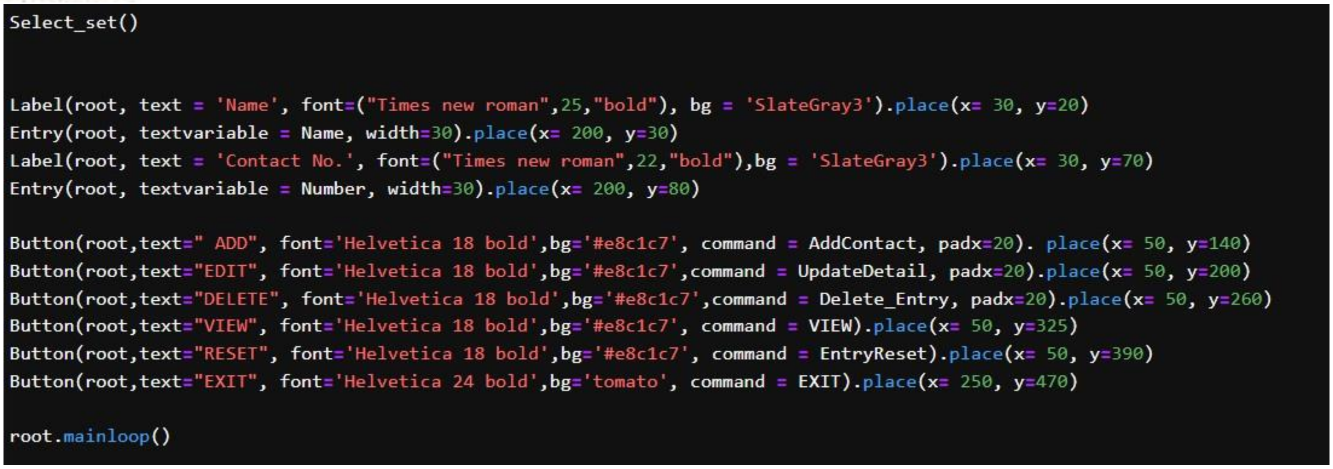
```
from tkinter import *
from tkinter import messagebox

#Initializing window
root = Tk()
root.geometry('700x550')
root.config(bg = '#d3f3f5')
root.title('Py Contact Book')
root.resizable(0,0)
contactlist = [
    ['Anjali','369854712'],
    ['Sakshi', '521155222'],
    ['Ruchika', '78945614'],
    ['Shivani', '58745246'],
    ['Suhani', '5846975'],
    ['Diksha', '5647892'],
    ['Varun', '89685320'],
    ['Tarun', '98564785'],
    ['Shikha','85967412']
    ]

Name = StringVar()
Number = StringVar()


#PythonGeeks - create frame
frame = Frame(root)
frame.pack(side = RIGHT)

scroll = Scrollbar(frame, orient=VERTICAL)
select = Listbox(frame, yscrollcommand=scroll.set,font=('Times new roman',16),bg="#f0fffc",width=20,height=20,borderwidth=3,relief="groove")
scroll.config (command=select.yview)
scroll.pack(side=RIGHT, fill=Y)
select.pack(side=LEFT,  fill=BOTH, expand=1)
```



```
def Selected():
    print("hello",len(select.curselection()))
    if len(select.curselection())==0:
        messagebox.showerror("Error", "Please Select the Name")
    else:
        return int(select.curselection()[0])

#PythonGeeks -function to add new contact
def AddContact():
    if Name.get()!="" and Number.get()!="":
        contactlist.append([Name.get() ,Number.get()])
        print(contactlist)
        Select_set()
        EntryReset()
        messagebox.showinfo("Confirmation", "Successfully Add New Contact")

    else:
        messagebox.showerror("Error","Please fill the information")


def UpdateDetail():
    if Name.get() and Number.get():
        contactlist[Selected()] = [Name.get(), Number.get()]


        messagebox.showinfo("Confirmation", "Successfully Update Contact")
        EntryReset()
        Select_set()

    elif not(Name.get()) and not(Number.get()) and not(len(select.curselection())==0):
        messagebox.showerror("Error", "Please fill the information")

    else:
        if len(select.curselection())==0:
            messagebox.showerror("Error", "Please Select the Name and \n press Load button")
```

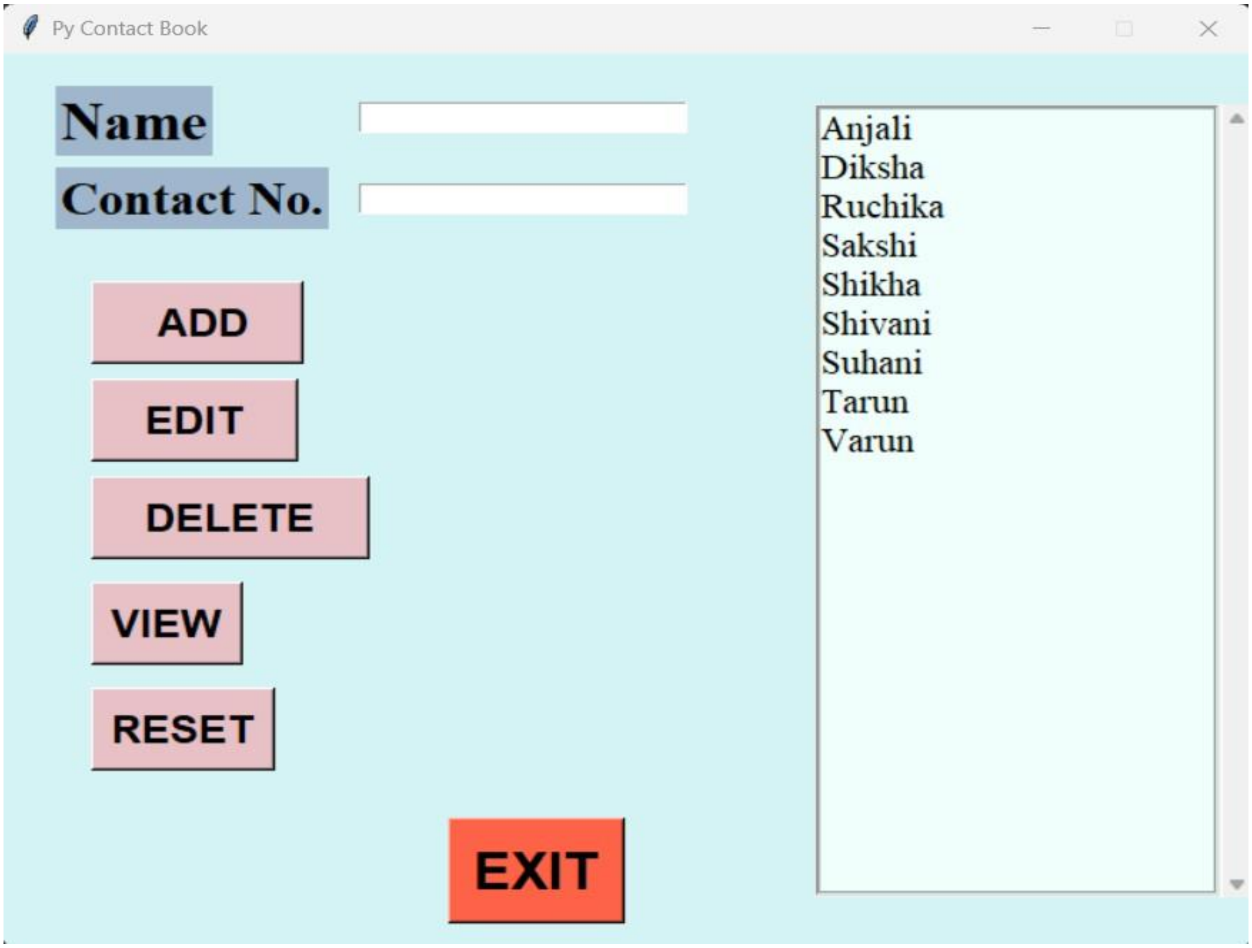


```
        else:
            message1 = """To Load the all information of \n
                          selected row press Load button\n.
                          """
            messagebox.showerror("Error", message1)

def EntryReset():
    Name.set('')
    Number.set('')

def Delete_Entry():
    if len(select.curselection())!=0:
        result=messagebox.askyesno('Confirmation','You Want to Delete Contact\n Which you selected')
        if result==True:
            del contactlist[Selected()]
            Select_set()
    else:
        messagebox.showerror("Error", 'Please select the Contact')


def VIEW():
    NAME, PHONE = contactlist[Selected()]
    Name.set(NAME)
    Number.set(PHONE)


def EXIT():
    root.destroy()


def Select_set() :
    contactlist.sort()
    select.delete(0,END)
    for name,phone in contactlist :
        select.insert (END, name)
```



```
Select_set()


Label(root, text = 'Name', font=("Times new roman",25,"bold"), bg = 'SlateGray3').place(x= 30, y=20)
Entry(root, textvariable = Name, width=30).place(x= 200, y=30)
Label(root, text = 'Contact No.', font=("Times new roman",22,"bold"),bg = 'SlateGray3').place(x= 30, y=70)
Entry(root, textvariable = Number, width=30).place(x= 200, y=80)

Button(root,text=" ADD", font='Helvetica 18 bold',bg='#e8c1c7', command = AddContact, padx=20). place(x= 50, y=140)
Button(root,text="EDIT", font='Helvetica 18 bold',bg='#e8c1c7',command = UpdateDetail, padx=20).place(x= 50, y=200)
Button(root,text="DELETE", font='Helvetica 18 bold',bg='#e8c1c7',command = Delete_Entry, padx=20).place(x= 50, y=260)
Button(root,text="VIEW", font='Helvetica 18 bold',bg='#e8c1c7', command = VIEW).place(x= 50, y=325)
Button(root,text="RESET", font='Helvetica 18 bold',bg='#e8c1c7', command = EntryReset).place(x= 50, y=390)
Button(root,text="EXIT", font='Helvetica 24 bold',bg='tomato', command = EXIT).place(x= 250, y=470)

root.mainloop()
```

1. Result/Output/Writing Summary:



```
Anjali
Diksha
Ruchika
Sakshi
Shikha
Shivani
Suhani
Tarun
Varun
```

Writing Summary:-

This project involves developing a contact book application that enhances user interaction through a graphical interface. Utilizing Python's Tkinter library, the application enables users to easily manage their contacts by adding new entries, updating existing ones, viewing details, and deleting contacts as needed. The implementation prioritizes usability by providing real-time feedback and validations. The project serves as an introduction to GUI programming and demonstrates practical application of data handling in Python.

4.Learning outcomes (What I have learnt):

- **Understanding Tkinter**:
- Gained proficiency in creating GUI applications using the Tkinter library, including layout design and widget management.
- **Event Handling**:
- Learnt how to handle user events (like button clicks and list selections) to create interactive applications.
- **Data Management**:
- Developed skills in managing and manipulating data structures (like lists) for real-time applications.