Report on

Joy of programming using Python

Submitted for Summer Internship Program

By : **Chiyam Kumar** (CS & IT, 1900270110017) and

Aryan Sharma (CS & IT, 1900270110013)

Under the Guidance of

Professor Dr. Pratima Singh and **Assistant Professor**

Mr. Binayak Parashar

AJAY KUMAR GARG ENGINEERING COLLEGE, GHAZIABAD

INDEX

PROJECT-6 (CONTACT BOOK)

SYSTEM REQUIREMENTS :

1. OPERATING SYSTEM:
    Windows 7 or above

2. RAM :
    4GB or more
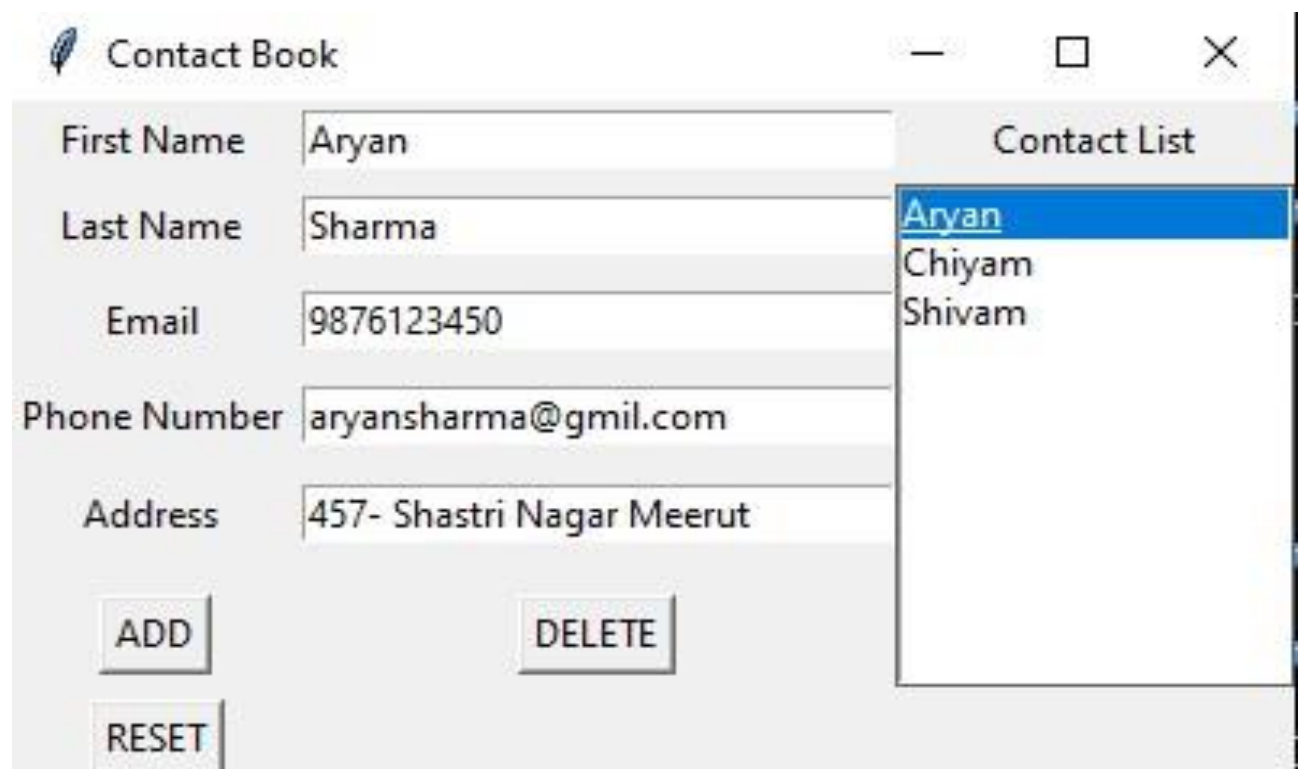
3. PYTHON version :
    3.70 or above

INPUT :

First we add the first name, last name, email, phone number, address and then we click on add button.

OUTPUT :

Contact details of the person can be showed by clicking on person's name.

Example:

CODE :

```python
import tkinter as tk
# from tkinter.ttk import *
class ContactBook(tk.Frame):
    def __init__(self,master=None):
        self.contacts = ContactList()
        self.fileName = "data.txt"
        self.reader = ContactFileReader(self.fileName)
        super().__init__(master)
        self.master = master
        self.pack()
        self.create_widgets()
        self.addListBox()
        self.fetch()
    def fetch(self):
        try:
            self.reader.open()
            z = self.reader.fetchAll()
            for item in z:
                self.contacts.addContact(item)
            self.reader.close()
            self.refreshList()
        except:
            print("Error")
    def writeFile(self):
        outfile = open(self.fileName,'w')
        if self.contacts.getContacts()!=None:
            for i in self.contacts.getContacts():
                outfile.write(i.first_name+","+i.last_name+","+i.email+","+i.phone_number+","+i.address)
                outfile.write('\n')
        outfile.close()
```

```python
def create_widgets(self):
    w = int(30)
    self.l1 = tk.Label(self)
    self.l2 = tk.Label(self)
    self.l3 = tk.Label(self)
    self.l4 = tk.Label(self)
    self.l5 = tk.Label(self)
    self.l1["text"] = "First Name "
    self.l2["text"] = "Last Name "
    self.l3["text"] = "Email "
    self.l4["text"] = "Phone Number "
    self.l5["text"] = "Address "
    self.l1.grid(row=0,column=0,pady=2)
    self.l2.grid(row=1,column=0,pady=2)
    self.l3.grid(row=2,column=0,pady=2)
    self.l4.grid(row=3,column=0,pady=2)
    self.l5.grid(row=4,column=0,pady=2)
    self.e1 = tk.Entry(self)
    self.e2 = tk.Entry(self)
    self.e3 = tk.Entry(self)
    self.e4 = tk.Entry(self)
    self.e5 = tk.Entry(self)
    self.e1.grid(row = 0, column = 1, pady = 2)
    self.e2.grid(row = 1, column = 1, pady = 2)
    self.e3.grid(row = 2, column = 1, pady = 2)
    self.e4.grid(row = 3, column = 1, pady = 2)
    self.e5.grid(row = 4, column = 1, pady = 2)
    self.e1["width"] = w
    self.e2["width"] = w
    self.e3["width"] = w
```

```python
            self.e4["width"] = w
            self.e5["width"] = w
            self.submit = tk.Button(self,command=self.addContact)
            self.submit["text"] = "ADD"
            self.submit.grid(row=6,column=0,pady=2)
            self.delete = tk.Button(self,command=self.delete)
            self.delete["text"] = "DELETE"
            self.delete.grid(row=6,column=1,pady=2)
            self.resetBtn = tk.Button(self,command=self.reset)
            self.resetBtn["text"] = "RESET"
            self.resetBtn.grid(row=7,column=0,pady=2)
    def selected(self,name):
            self.reset()
            k = None
            for i in self.contacts.getContacts():
                if name==i.getName():
                    k = i
            if k==None:
                print("Not Found")
                return
            self.e1.insert(tk.END,k.getName())
            self.e2.insert(tk.END,k.getLastName())
            self.e3.insert(tk.END,k.getEmail())
            self.e4.insert(tk.END,k.getPhone())
            self.e5.insert(tk.END,k.getAddress())
    def CurSelect(self,event):
            widget = event.widget
            selection=widget.curselection()
            picked = widget.get(selection)
            self.selected(picked)
```

```python
    def addListBox(self):
        self.l9 = tk.Label(self)
        self.l9["text"] = "Contact List"
        self.l9.grid(row=0,column=3,pady=2)
        self.listbox = tk.Listbox(self)
        self.listbox.bind('<<ListboxSelect>>',self.CurSelect)
        self.listbox.bind('<1>',self.list_click)
        self.listbox.grid(row=1,column=3,rowspan=6,pady=2)
    def reset(self):
        self.e1.delete(0,tk.END)
        self.e2.delete(0,tk.END)
        self.e3.delete(0,tk.END)
        self.e4.delete(0,tk.END)
        self.e5.delete(0,tk.END)
    def list_click(self,event):
        w = event.widget
        index = w.nearest(event.y)
        w._selection = index
    def refreshList(self):
        self.listbox.delete(0,tk.END)
        z = [i.getName() for i in self.contacts.getContacts()]
        for item in z:
            self.listbox.insert(tk.END,item)
    def addContact(self):
        name = self.e1.get()
        lastName = self.e2.get()
        email = self.e3.get()
        phone = self.e4.get()
        address = self.e5.get()
        c = Contact(name,lastName,email,phone,address)
```

```python
            self.contacts.addContact(c)
            self.refreshList()
            self.writeFile()
            self.reset()
        def delete(self):
            self.contacts.deleteContact(self.e1.get())
            self.writeFile()
            self.refreshList()
            self.reset()


class ContactList:
    def __init__(self):
        self._conList = list()
    def addContact(self,contact):
        self._conList.append(contact)
    def getContacts(self):
        k = list()
        for i in self._conList:
            k.append(i)
        return k
    def deleteContact(self,name):
        index = None
        for i in range(len(self._conList)):
            if self._conList[i].getName()==name:
                index = i
                break
        if not index==None:
            self._conList.pop(index)
class Contact:
    def __init__(self,name="",lastName="",phone="",email="",address=""):
```

```python
            self.first_name = name
            self.last_name = lastName
            self.phone_number = phone
            self.email = email
            self.address = address
    def printContact(self):
        print(self.first_name)
        print(self.last_name)
        print(self.phone_number)
        print(self.email)
        print(self.address)
    def getName(self):
        return self.first_name
    def getLastName(self):
        return self.last_name
    def getEmail(self):
        return self.email
    def getAddress(self):
        return self.address
    def getPhone(self):
        return self.phone_number
    def set_name(self,name):
        self.first_name = name
    def setLastName(self,name):
        self.last_name = name
    def setEmail(self,email):
        self.email = email
    def setPhone(self,phone):
        self.phone_number = phone
    def setAddress(self,addr):
```

```python
181            self.address = addr
182
183    class ContactFileReader:
184        def __init__(self,inputSrc):
185            self._inputSrc = inputSrc
186            self._inputFile = None
187        def open(self):
188            self._inputFile = open(self._inputSrc,'r')
189        def close(self):
190            self._inputFile.close()
191        def fetchAll(self):
192            k = list()
193            contact = self.fetchRecord()
194            while contact!=None:
195                k.append(contact)
196                contact = self.fetchRecord()
197            return k
198        def fetchRecord(self):
199            line = self._inputFile.readline()
200            if line=="":
201                return None
202            k = line.split(',')
203            contact = Contact()
204            contact.first_name = k[0]
205            contact.last_name = k[1]
206            contact.email = k[2]
207            contact.phone_number = k[3]
208            contact.address = k[4]
209            return contact
```
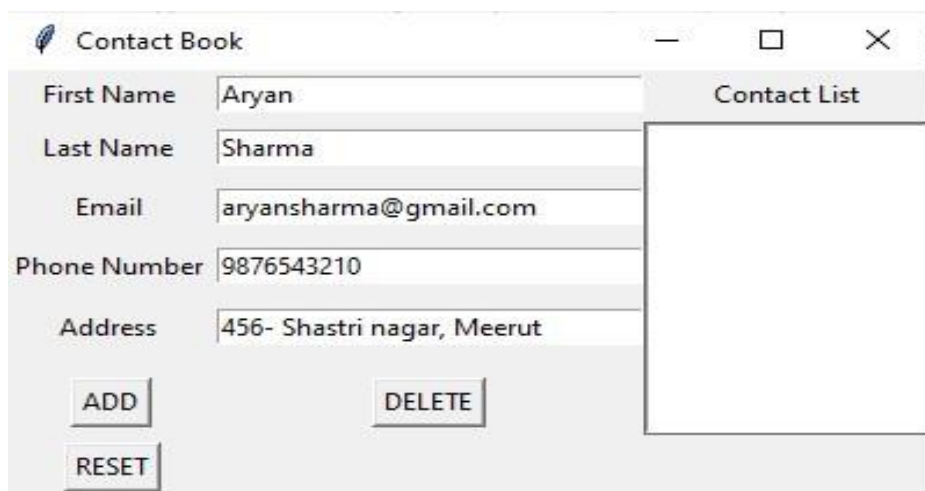
```python
210
211
212    root = tk.Tk()
213    root.title("Contact Book")
214    app = ContactBook(master=root)
215    app.mainloop()
```

TEST CASES :

1. To add a contact in the contact list.

Step 1:

Fill all the details of the person whose contact you are adding which includes first name, last name, email, phone number and address.
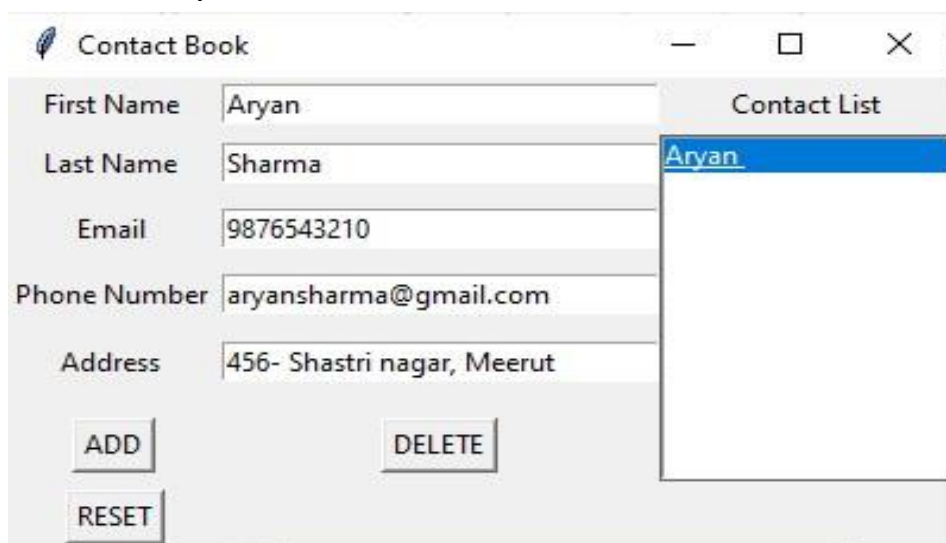
Step 2:

Then click on add button and your contact will be added successfully.
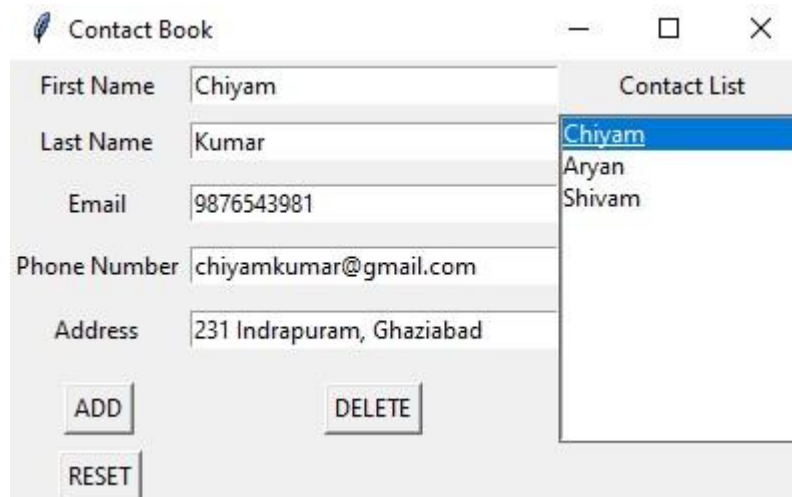
2. To delete a contact from contact list.

Step 1:
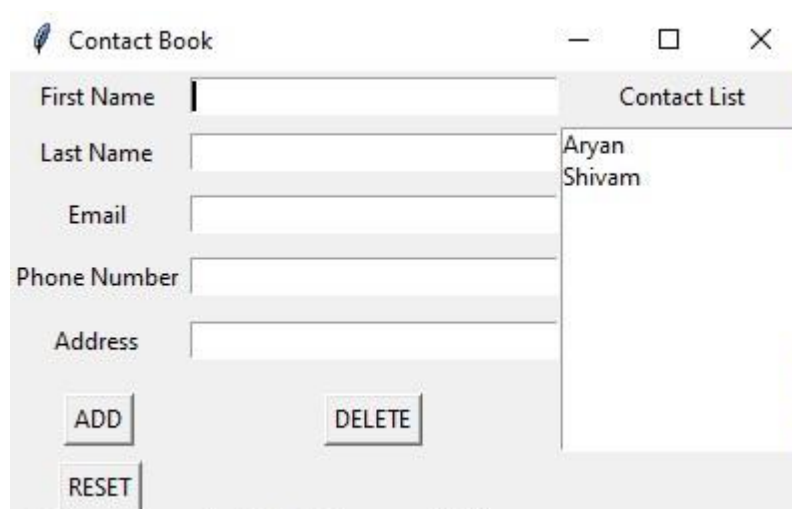
Click on the contact from contact list which you want to delete.

Step 2:

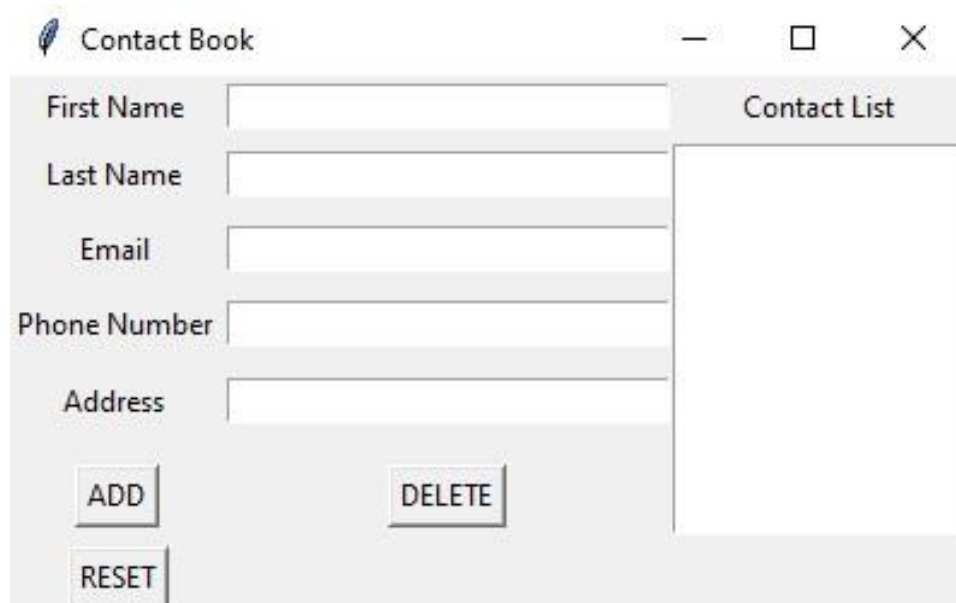Then click on the delete option below the contact details and your contact will be deleted successfully.

3. To reset the space for filling contact info.

Step 1: Click on the reset option to reset the contact info.

Step 2: Then the contact info is rest successfully.

ASSIGNMENT

1. Write a Python program to take input of a string and reverse the string.

PROGRAM:

```
def reversestring(s) :
    print("reversed string is :",s[::-1])
s=input("enter the string")
print("entered string is :",s)
reversestring(s)
|
```

OUTPUT:

```
enter the string chiyam
entered string is :  chiyam
reversed string is : mayihc
```

2. Write a python program to check a list is empty or not.

```
a = []
if len(a) == 0 :
    print("list is empty")
else :
    print("list is not empty ")
```

OUTPUT:

```
list is empty
>>>
```

3. Write a python function that take two lists and returns True I
   they have at least one common member.

PROGRAM:

```
def common_data(list1,list2):
    result = False
    for x in list1:
        for y in list2:
            if x==y :
                result = True
                return result
print(common_data([2,3,4],[4,5,6]))
print(common_data([5,6,7],[8,9,10]))
```

OUTPUT :

```
True
None
>>> |
```