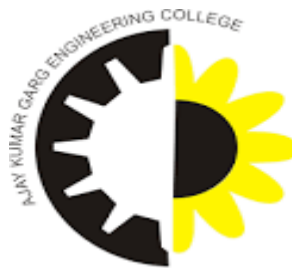


Report on
Joy of programming using Python

Submitted for Summer Internship Program

By : 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
YEAR 2020-21

INDEX

1.	PROJECT -6
2.	SYSTEM REQUIREMENTS
3.	INPUT
4.	OUTPUT
5.	EXAMPLE
6.	CODE
7.	TEST CASE
8.	ASSIGNMENT

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

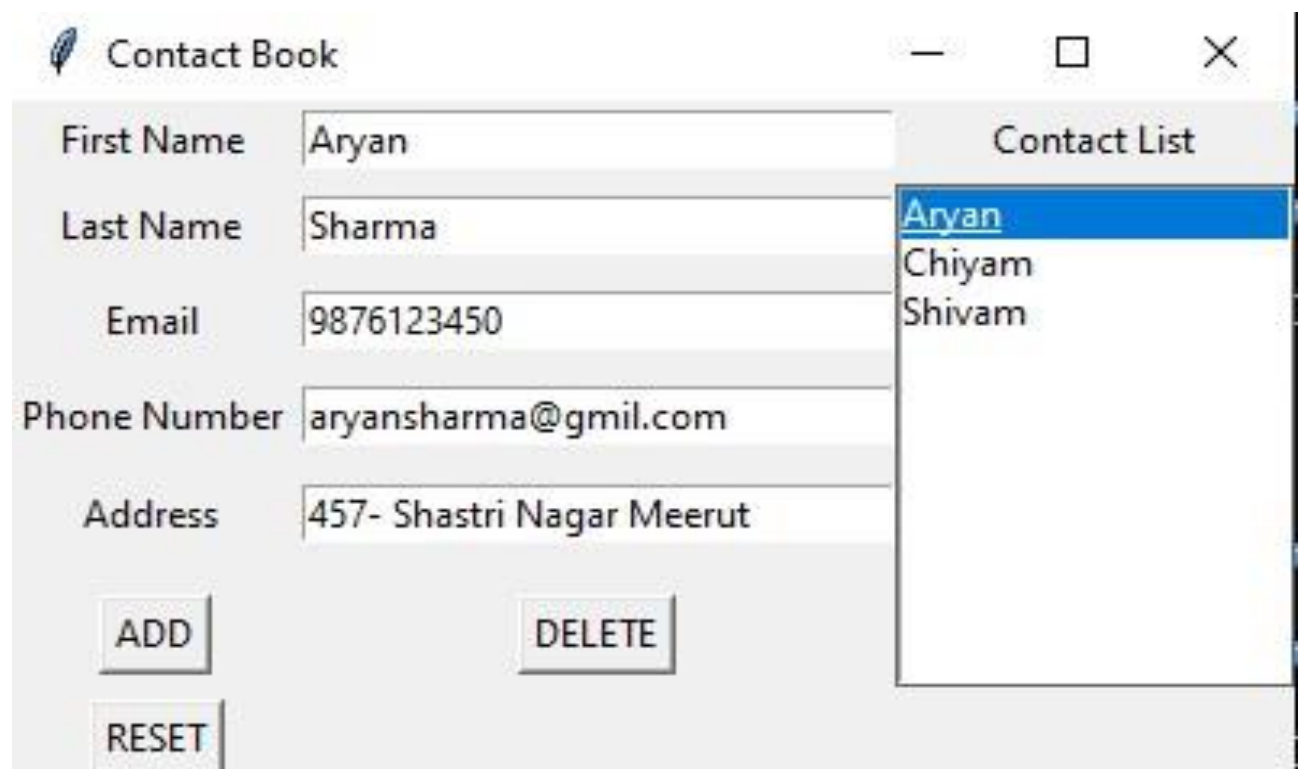
INPUT :

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:



The screenshot displays a web application titled "Contact Book". On the left, there is a form with the following fields: "First Name" (Aryan), "Last Name" (Sharma), "Email" (9876123450), "Phone Number" (aryansharma@gmail.com), and "Address" (457- Shastri Nagar Meerut). Below these fields are three buttons: "ADD", "DELETE", and "RESET". On the right, a "Contact List" panel shows a list of names: "Aryan" (highlighted in blue), "Chiyam", and "Shivam". The application window includes standard OS controls (minimize, maximize, close) in the top right corner.

Contact Book		Contact List
First Name	Aryan	Aryan
Last Name	Sharma	Chiyam
Email	9876123450	Shivam
Phone Number	aryansharma@gmail.com	
Address	457- Shastri Nagar Meerut	

Buttons: ADD, DELETE, RESET

CODE :

```
1  import tkinter as tk
2  # from tkinter.ttk import *
3  class ContactBook(tk.Frame):
4      def __init__(self, master=None):
5          self.contacts = ContactList()
6          self.fileName = "data.txt"
7          self.reader = ContactFileReader(self.fileName)
8          super().__init__(master)
9          self.master = master
10         self.pack()
11         self.create_widgets()
12         self.addListBox()
13         self.fetch()
14     def fetch(self):
15         try:
16             self.reader.open()
17             z = self.reader.fetchAll()
18             for item in z:
19                 self.contacts.addContact(item)
20             self.reader.close()
21             self.refreshList()
22         except:
23             print("Error")
24     def writeFile(self):
25         outfile = open(self.fileName, 'w')
26         if self.contacts.getContacts() != None:
27             for i in self.contacts.getContacts():
28                 outfile.write(i.first_name+" "+i.last_name+" "+i.email+" "+i.phone_number+" "+i.address)
29                 outfile.write('\n')
30         outfile.close()
```

```
31     def create_widgets(self):
32         w = int(30)
33         self.l1 = tk.Label(self)
34         self.l2 = tk.Label(self)
35         self.l3 = tk.Label(self)
36         self.l4 = tk.Label(self)
37         self.l5 = tk.Label(self)
38         self.l1["text"] = "First Name "
39         self.l2["text"] = "Last Name "
40         self.l3["text"] = "Email "
41         self.l4["text"] = "Phone Number "
42         self.l5["text"] = "Address "
43         self.l1.grid(row=0, column=0, pady=2)
44         self.l2.grid(row=1, column=0, pady=2)
45         self.l3.grid(row=2, column=0, pady=2)
46         self.l4.grid(row=3, column=0, pady=2)
47         self.l5.grid(row=4, column=0, pady=2)
48         self.e1 = tk.Entry(self)
49         self.e2 = tk.Entry(self)
50         self.e3 = tk.Entry(self)
51         self.e4 = tk.Entry(self)
52         self.e5 = tk.Entry(self)
53         self.e1.grid(row = 0, column = 1, pady = 2)
54         self.e2.grid(row = 1, column = 1, pady = 2)
55         self.e3.grid(row = 2, column = 1, pady = 2)
56         self.e4.grid(row = 3, column = 1, pady = 2)
57         self.e5.grid(row = 4, column = 1, pady = 2)
58         self.e1["width"] = w
59         self.e2["width"] = w
60         self.e3["width"] = w
```

```

61 |         self.e4["width"] = w
62 |         self.e5["width"] = w
63 |         self.submit = tk.Button(self,command=self.addContact)
64 |         self.submit["text"] = "ADD"
65 |         self.submit.grid(row=6,column=0,pady=2)
66 |         self.delete = tk.Button(self,command=self.delete)
67 |         self.delete["text"] = "DELETE"
68 |         self.delete.grid(row=6,column=1,pady=2)
69 |         self.resetBtn = tk.Button(self,command=self.reset)
70 |         self.resetBtn["text"] = "RESET"
71 |         self.resetBtn.grid(row=7,column=0,pady=2)
72 |     def selected(self,name):
73 |         self.reset()
74 |         k = None
75 |         for i in self.contacts.getContacts():
76 |             if name==i.getName():
77 |                 k = i
78 |         if k==None:
79 |             print("Not Found")
80 |             return
81 |         self.e1.insert(tk.END,k.getName())
82 |         self.e2.insert(tk.END,k.getLastName())
83 |         self.e3.insert(tk.END,k.getEmail())
84 |         self.e4.insert(tk.END,k.getPhone())
85 |         self.e5.insert(tk.END,k.getAddress())
86 |     def CurSelect(self,event):
87 |         widget = event.widget
88 |         selection=widget.curselection()
89 |         picked = widget.get(selection)
90 |         self.selected(picked)

```

```
91     def addListBox(self):
92         self.l9 = tk.Label(self)
93         self.l9["text"] = "Contact List"
94         self.l9.grid(row=0, column=3, pady=2)
95         self.listbox = tk.Listbox(self)
96         self.listbox.bind('<<ListboxSelect>>', self.CurSelect)
97         self.listbox.bind('<1>', self.list_click)
98         self.listbox.grid(row=1, column=3, rowspan=6, pady=2)
99     def reset(self):
100         self.e1.delete(0, tk.END)
101         self.e2.delete(0, tk.END)
102         self.e3.delete(0, tk.END)
103         self.e4.delete(0, tk.END)
104         self.e5.delete(0, tk.END)
105     def list_click(self, event):
106         w = event.widget
107         index = w.nearest(event.y)
108         w._selection = index
109     def refreshList(self):
110         self.listbox.delete(0, tk.END)
111         z = [i.getName() for i in self.contacts.getContacts()]
112         for item in z:
113             self.listbox.insert(tk.END, item)
114     def addContact(self):
115         name = self.e1.get()
116         lastName = self.e2.get()
117         email = self.e3.get()
118         phone = self.e4.get()
119         address = self.e5.get()
120         c = Contact(name, lastName, email, phone, address)
```



```
121         self.contacts.addContact(c)
122         self.refreshList()
123         self.writeFile()
124         self.reset()
125     def delete(self):
126         self.contacts.deleteContact(self.e1.get())
127         self.writeFile()
128         self.refreshList()
129         self.reset()
130
131     class ContactList:
132     def __init__(self):
133         self._conList = list()
134     def addContact(self,contact):
135         self._conList.append(contact)
136     def getContacts(self):
137         k = list()
138         for i in self._conList:
139             k.append(i)
140         return k
141     def deleteContact(self,name):
142         index = None
143         for i in range(len(self._conList)):
144             if self._conList[i].getName()==name:
145                 index = i
146                 break
147         if not index==None:
148             self._conList.pop(index)
149     class Contact:
150     def __init__(self,name="",lastName="",phone="",email="",address=""):
```

```
151         self.first_name = name
152         self.last_name = lastName
153         self.phone_number = phone
154         self.email = email
155         self.address = address
156     def printContact(self):
157         print(self.first_name)
158         print(self.last_name)
159         print(self.phone_number)
160         print(self.email)
161         print(self.address)
162     def getName(self):
163         return self.first_name
164     def getLastName(self):
165         return self.last_name
166     def getEmail(self):
167         return self.email
168     def getAddress(self):
169         return self.address
170     def getPhone(self):
171         return self.phone_number
172     def set_name(self, name):
173         self.first_name = name
174     def setLastName(self, name):
175         self.last_name = name
176     def setEmail(self, email):
177         self.email = email
178     def setPhone(self, phone):
179         self.phone_number = phone
180     def setAddress(self, addr):
```

```
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
```

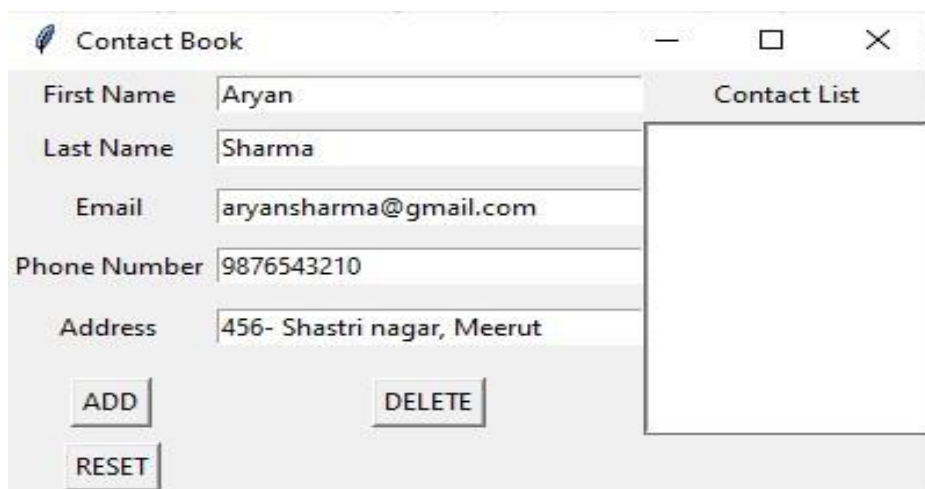
```
210
211
212     root = tk.Tk()
213     root.title("Contact Book")
214     app = ContactBook(master=root)
215     app.mainloop()
```

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.



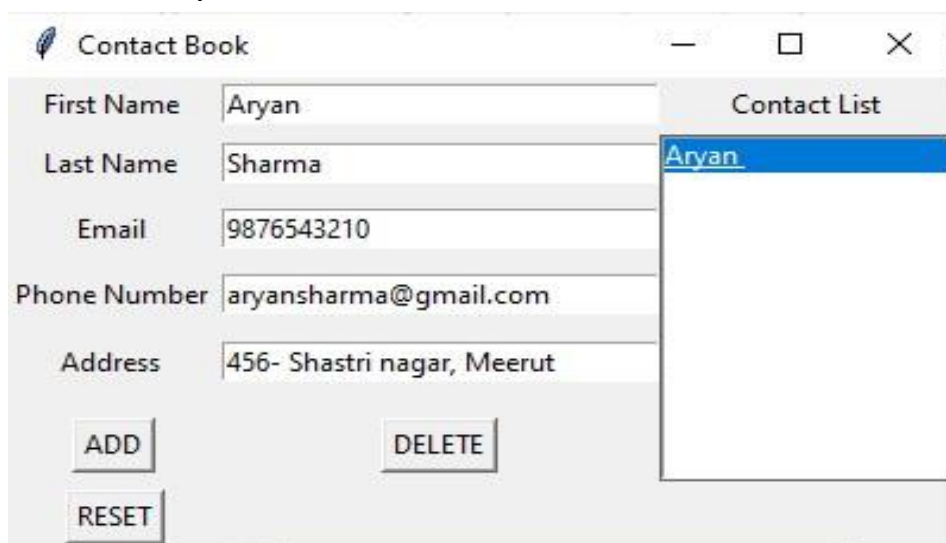
The screenshot shows a window titled "Contact Book" with a standard Windows title bar (minimize, maximize, close buttons). On the left, there is a form with the following fields and values:

Field	Value
First Name	Aryan
Last Name	Sharma
Email	aryansharma@gmail.com
Phone Number	9876543210
Address	456- Shastri nagar, Meerut

Below the form are three buttons: "ADD", "DELETE", and "RESET". To the right of the form is a panel titled "Contact List" which is currently empty.

Step 2:

Then click on add button and your contact will be added successfully.



The screenshot shows the same "Contact Book" window. The form fields now contain the following values:

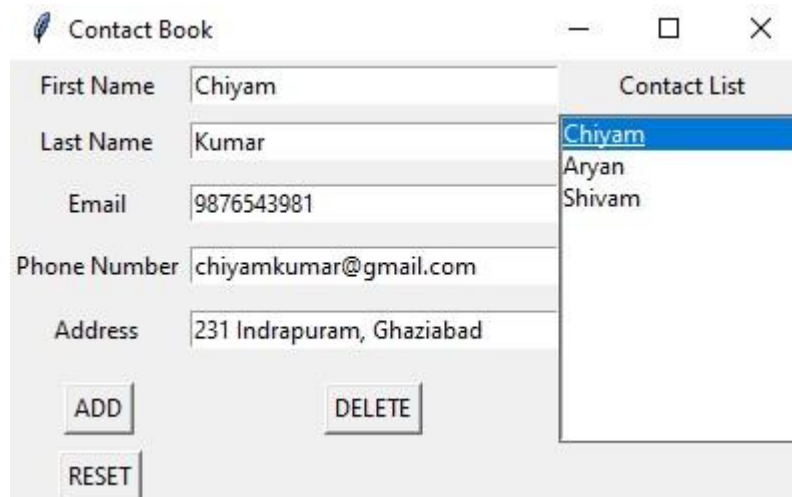
Field	Value
First Name	Aryan
Last Name	Sharma
Email	9876543210
Phone Number	aryansharma@gmail.com
Address	456- Shastri nagar, Meerut

The "ADD" button is still present. The "Contact List" panel on the right now contains one entry, "Aryan", which is highlighted with a blue background.

2. To delete a contact from contact list.

Step 1:

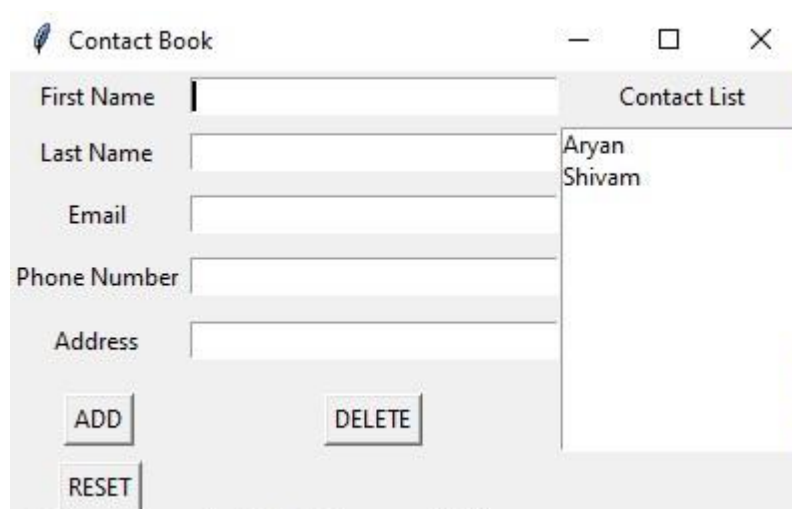
Click on the contact from contact list which you want to delete.



The screenshot shows a window titled "Contact Book" with a standard Windows title bar (minimize, maximize, close buttons). On the left is a form with the following fields: First Name (Chiyam), Last Name (Kumar), Email (9876543981), Phone Number (chiyamkumar@gmail.com), and Address (231 Indrapuram, Ghaziabad). Below the form are three buttons: ADD, DELETE, and RESET. On the right is a "Contact List" panel. It has a header "Contact List" and a list of three contacts: "Chiyam" (highlighted in blue), "Aryan", and "Shivam".

Step 2:

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



The screenshot shows the same "Contact Book" window. The form fields on the left are now empty. The "DELETE" button is still present. The "Contact List" panel on the right now only contains two contacts: "Aryan" and "Shivam". The "Chiyam" contact has been removed from the list.

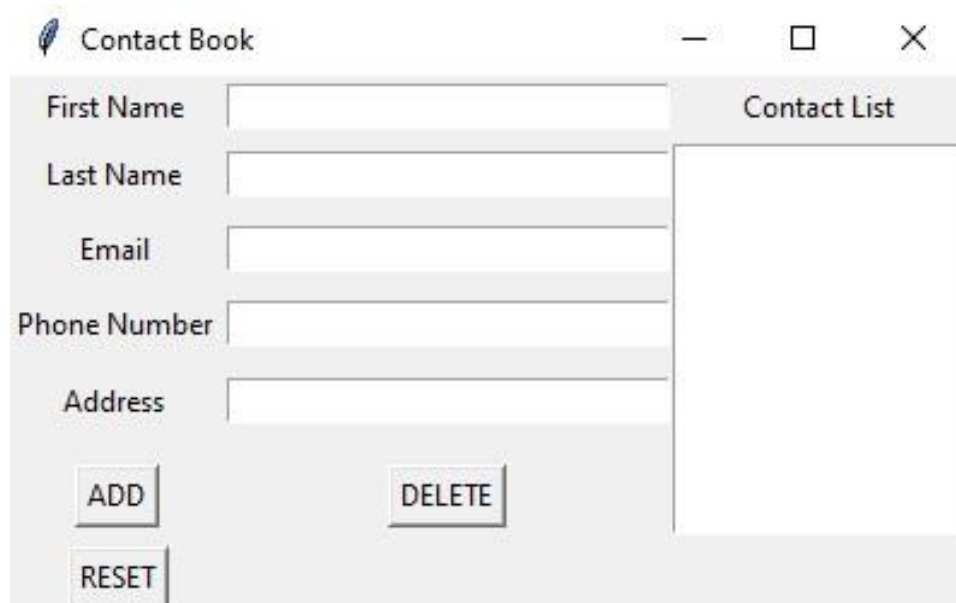
3. To reset the space for filling contact info.

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



The image shows a window titled "Contact Book" with standard window controls (minimize, maximize, close). On the left is a form with the following fields: "First Name" (Aryan), "Last Name" (Sharma), "Email" (abc@gmail.com), "Phone Number" (9876543210), and "Address" (231 indrapuram). Below the form are three buttons: "ADD", "DELETE", and "RESET". On the right is a panel titled "Contact List" which is currently empty.

Step 2: Then the contact info is rest successfully.



The image shows the same "Contact Book" window, but the form fields are now empty. The "First Name", "Last Name", "Email", "Phone Number", and "Address" fields are all blank. The "ADD", "DELETE", and "RESET" buttons are still present at the bottom. The "Contact List" panel on the right remains empty.

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.

PROGRAM:

```
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 if 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  
>>> |
```