

CONTACT MANAGEMENT SYSTEM

END-TERM REPORT

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By:

<i>S.no.</i>	<i>Name</i>	<i>Roll No.</i>	<i>Registration no.</i>
<i>1.</i>	<i>AKHIL NEELAM</i>	<i>01</i>	<i>12001936</i>

Courses Code: INT213



School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

Student Declaration

This is to declare that this report has been written by me. No part of the report is copied from other sources. All information included from other sources have been duly acknowledged. I/We aver that if any part of the report is found to be copied, I/we are shall take full responsibility for it.

Signature:

Name: AKHIL NEELAM

Roll Number:RK19QWA01

Date:30-10-2020

TABLE OF CONTENTS

TITLE	PAGENO
Background and objectives of project assigned	5
Introduction of Project	6
Module Description	7
Screenshots	8
Source Code	14
Results	24

BONAFIDE CERTIFICATE

Certified that this project report “**Contact Management System**” is the bonafide work of “AKHIL NEELAM” who carried out the project work under my supervision.

<<Signature of the Supervisor>>(Due to Covid 19, signature is exempted)

<<Name of supervisor>>

<<Academic Designation>>

<<ID of Supervisor>>

<<Department of
Supervisor>>

Objective

The primary objective of this project is to implement what we've learnt throughout our course of Python programming and use that to develop a Graphical User Interface (GUI) for Contact management system with all the required functionalities. This project also aims at Business Process automation by providing a user friendly interface to the users to let them easily use the contact book. Through this project we aim to target the easy to understand by the user and operator and good user interface.

This is a simple GUI based mission that's very clean to apprehend and use. Talking about the device, it includes all the required features which encompass including, viewing, deleting and updating contact lists. While including the contact of a person, he/she has to provide first name, closing name, gender, deal with and call info. The user can automatically update the touch list if he/she wishes to. For this, the person has to double click on the a file that he/she wishes to edit. There's an external database connection file used in this mini project to save user's data permanently.

Introduction

The “contact Management System “ has been developed to override the problems prevailing in the practice manual system . This software is supposed to eliminate and in some cases reduce the hardships faced by this existing system. Moreover this system is designed for the particular need of the company to carry out operations in a smooth and effective manner

The application is reduced as much as possible to avoid errors while entering the data . It also provides error message while entering invalid data . No formal knowledge is needed for the user to use this system. Thus by this all it proves it is user friendly .contact management system , as described above , can assist the user to concentrate on their other activities rather to concentrate on the record keeping .Thus it will help organization in better utilization of resources.

Every organization ,whether big or small , has challenges to overcome and managing the information of contact , profile, Mobile ,Emails etc. Every contact management system has different contact needs, therefore we design exclusive employee management systems that are adapted to your managerial requirements .These systems will ultimately allow you to better manage resources .

Modules of Contact Management System

Add New Contact: This module is to manage the contact details such as First Name, Last Name, Gender, Age ,Home Address, Phone Number, Religious, and even Nationality

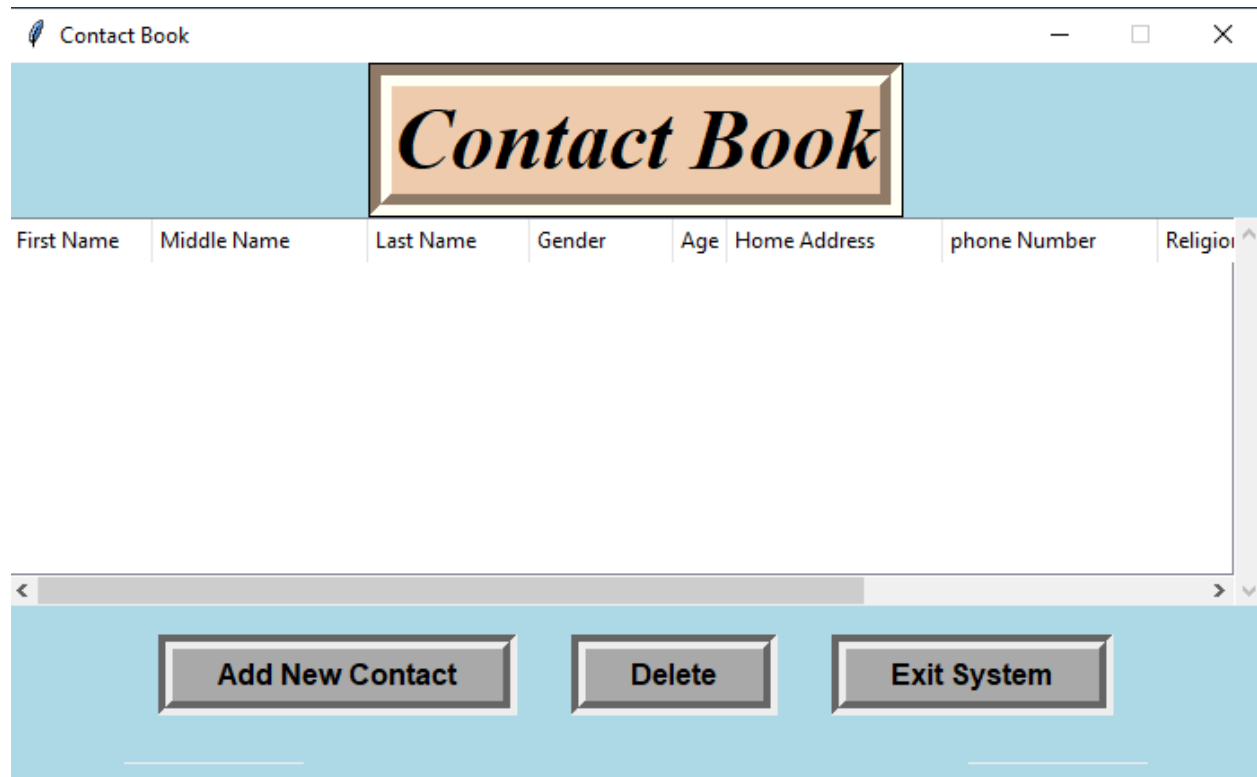
Delete: This Module is to Delete the unwanted contact from database

Edit the existing contact: This module is to Edit the contact which are updated at database by just double clicking contact

Exit the system: This Module is to exit from the contact management System

GUI Screenshots:

1.WELCOME SCREEN:



2.Add New Contact (with input values)

Contact Details

Adding New Contacts

First Name	Ramesh
Middle Name	me
Last Name	Ram
Gender	<input checked="" type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Transgender
Age	30
Home Address	ri Nagar, Almasguda,HYI
Phone Number	9502741527
Religion	Hindu
Nationality	Indian

Save

3. Save of the Contact (with input values)

First Name	Middle Name	Last Name	Gender	Age	Home Address	phone Number	Religion
Ramesh	me	Ram	Male	30	House no 14-25/A,sh	9502741527	Hindu

Buttons: Add New Contact, Delete, Exit System

4. Save in the database:

id	first_name	middle_name	last_name	gender	age	home_address	phone_number	religion	nationality
1	8 Ramesh	me	Ram	Male	30	House no 14-25/A,shesh...	9502741527	Hindu	Indian

5. Updating or Edit the Contact *(on double clicking the existing contact)*

The screenshot shows a window titled 'Contact Details' with a blue 'Updating Contacts' button at the top. Below the button is a form with the following fields and values:

First Name	Ramesh
Middle Name	me
Last Name	Shyam
Gender	<input checked="" type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Transgender
Age	31
Home Address	House no 14-25/A,shesh
Phone Number	9502741528
Religion	Hindu
Nationality	Indian

At the bottom of the form is a blue 'Update' button.

Delete the existing contact: *(on clicking the contact which want to delete click yes to delete the contact)*

The screenshot shows a window titled 'Contact Book' with a table of contacts. A confirmation dialog box is open in the foreground, asking 'Are You Sure You Want to Delete This Record?' with 'Yes' and 'No' buttons. The 'Delete' button in the bottom toolbar is highlighted with a yellow border.

First Name	Middle Name	Last Name	Gender	Age	Home Address	phone Number	Religion
Ramesh	me	Shyam	Male	31	House no 14-25/A,sh	9502741528	Hindu

Buttons at the bottom: Add New Contact, Delete, Exit System.

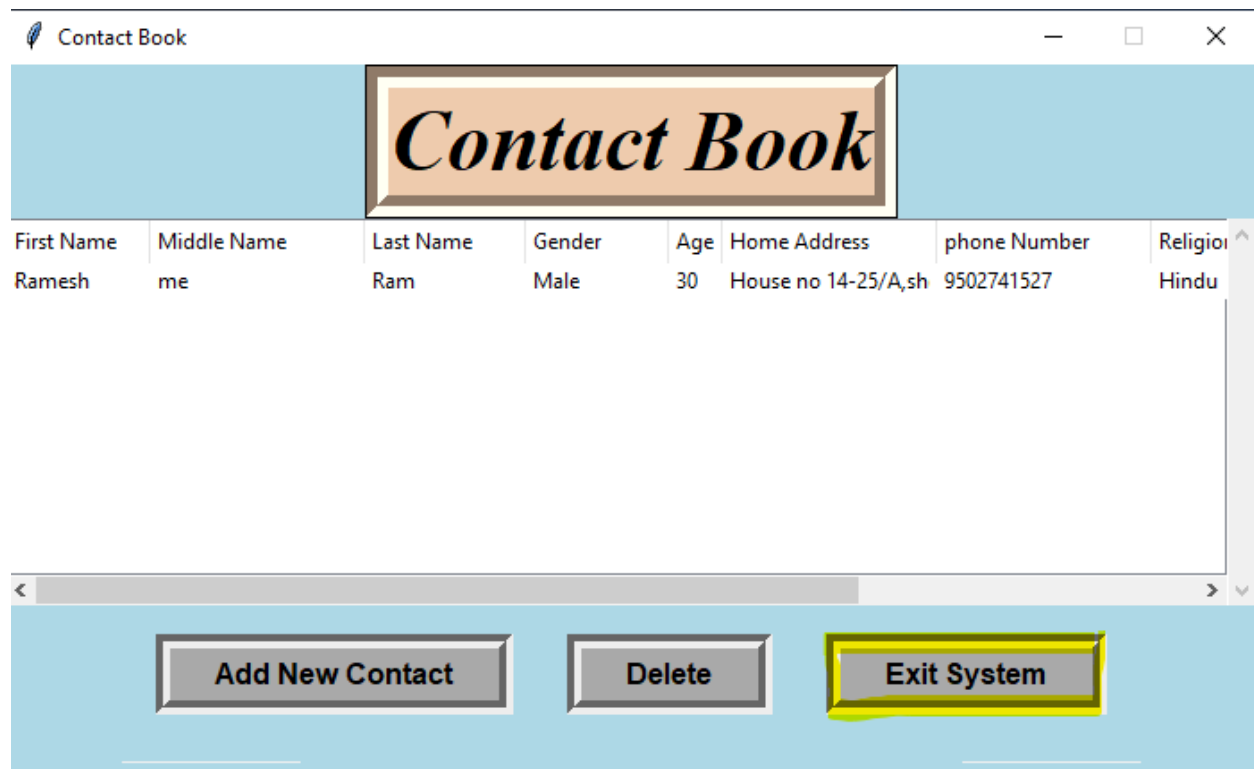
6. Updated contact Details *(Here Last name, Phone number, age is updated)*

First Name	Middle Name	Last Name	Gender	Age	Home Address	phone Number	Religion
Ramesh	me	Shyam	Male	31	House no 14-25/A,sh	9502741528	Hindu

7. View in the Database *(After Update or edit the contact)*

	id	first_name	middle_name	last_name	gender	age	home_address	phone_number	religion	nationality
1	8	Ramesh	me	Shyam	Male	31	House no 14-25/A,shesh...	9502741528	Hindu	Indian

8. Exit System *(on clicking Exit system)*

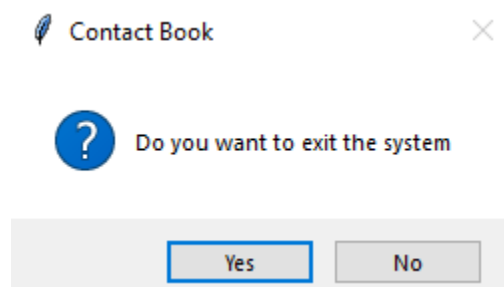


The screenshot shows a window titled "Contact Book" with a light blue header and footer. The header contains a stylized "Contact Book" title. The main area displays a table with contact information. The footer contains three buttons: "Add New Contact", "Delete", and "Exit System". The "Exit System" button is highlighted with a yellow border.

First Name	Middle Name	Last Name	Gender	Age	Home Address	phone Number	Religion
Ramesh	me	Ram	Male	30	House no 14-25/A,sh	9502741527	Hindu

Buttons: Add New Contact, Delete, Exit System

(Click Yes to exit from the system)



The screenshot shows a window titled "Contact Book" with a light blue header and footer. The header contains a stylized "Contact Book" title. The main area displays a confirmation dialog box with a question mark icon and the text "Do you want to exit the system". Below the text are two buttons: "Yes" and "No". The "Yes" button is highlighted with a blue border.

Do you want to exit the system

Buttons: Yes, No

Source Code

```
from tkinter import *
3  import sqlite3
4  import tkinter
5  import tkinter.ttk as ttk
6  import tkinter.messagebox as tkMessageBox
7
8
9  root = Tk()
10 root.title("Contact Book")
11 root.geometry("700x400+0+0")
12 root.resizable(0, 0)
13 root.config(bg="light blue")
14
15 # VARIABLES
16 f_name = StringVar()
17 m_name = StringVar()
18 l_name = StringVar()
19 age = StringVar()
20 home_address = StringVar()
21 phone_number = StringVar()
22 gender = StringVar()
23 religion = StringVar()
24 nationality = StringVar()
25
26 # METHODS
27 def Exit():
28     wayOut = tkinter.messagebox.askyesno("Contact Book", "Do you want to exit the
system")
29     if wayOut > 0:
30         root.destroy()
31         return
32
33 def Reset():
34     f_name.set("")
35     m_name.set("")
36     l_name.set("")
37     gender.set("")
38     age.set("")
39     home_address.set("")
40     phone_number.set("")
41     religion.set("")
42     nationality.set("")
43
44 def Database():
45     conn = sqlite3.connect("contactdb.db")
```

```

46     cursor = conn.cursor()
47     cursor.execute(
48         "CREATE TABLE IF NOT EXISTS `contactable` (id INTEGER NOT NULL
PRIMARY KEY AUTOINCREMENT, first_name TEXT, middle_name TEXT, last_name
TEXT, gender TEXT, age TEXT, home_address TEXT, phone_number TEXT, religion
TEXT, nationality TEXT)")
49     cursor.execute("SELECT * FROM `contactable` ORDER BY `last_name` ASC")
50     fetch = cursor.fetchall()
51     for data in fetch:
52         tree.insert('', 'end', values=(data))
53     cursor.close()
54     conn.close()
55
56
57     def Submit():
58         if f_name.get() == "" or m_name.get() == "" or l_name.get() == "" or gender.get()
== "" or age.get() == "" or home_address.get() == "" or phone_number.get() == "" or
religion.get() == "" or nationality.get() == "" or phone_number.get() == str:
59             result = tkMessageBox.showwarning('', 'Please Complete The Required Field',
icon="warning")
60         else:
61             tree.delete(*tree.get_children())
62             conn = sqlite3.connect("contactdb.db")
63             cursor = conn.cursor()
64             cursor.execute(
65                 "INSERT INTO `contactable` (first_name, middle_name, last_name, gender,
age, home_address, phone_number, religion, nationality) VALUES(?, ?, ?, ?, ?, ?, ?, ?, ?)",
(str(f_name.get()), str(m_name.get()), str(l_name.get()), str(gender.get()), int(age.get()),
str(home_address.get()),
66                 int(phone_number.get()), str(religion.get()), str(nationality.get())))
67             conn.commit()
68             cursor.execute("SELECT * FROM `contactable` ORDER BY `last_name` ASC")
69             fetch = cursor.fetchall()
70             for data in fetch:
71                 tree.insert('', 'end', values=(data))
72             cursor.close()
73             conn.close()
74             f_name.set("")
75             m_name.set("")
76             l_name.set("")
77             gender.set("")
78             age.set("")
79             home_address.set("")
80             phone_number.set("")
81             religion.set("")
82             nationality.set("")
83
84
85     def Update():

```

```

86     if gender.get() == "" or phone_number.get() == str:
87         result = tkMessageBox.showwarning('Please Complete The Required Field ',
88         icon="warning")
89     else:
90         tree.delete(*tree.get_children())
91         conn = sqlite3.connect("contactdb.db")
92         cursor = conn.cursor()
93         cursor.execute(
94             "UPDATE `contactable` SET `first_name` = ?, `middle_name` = ? , `last_name`
95             = ?, `gender` = ?, `age` = ?, `home_address` = ?, `phone_number` = ?, `religion` = ?,
96             `nationality` = ? WHERE `id` = ?",
97             (str(f_name.get()), str(m_name.get()), str(l_name.get()), str(gender.get()),
98             int(age.get()), str(home_address.get()),
99             int(phone_number.get()), str(religion.get()), str(nationality.get()), int(id)))
100        conn.commit()
101        cursor.execute("SELECT * FROM `contactable` ORDER BY `last_name` ASC")
102        fetch = cursor.fetchall()
103        for data in fetch:
104            tree.insert('', 'end', values=(data))
105        cursor.close()
106        conn.close()
107        f_name.set("")
108        m_name.set("")
109        l_name.set("")
110        gender.set("")
111        age.set("")
112        home_address.set("")
113        phone_number.set("")
114        religion.set("")
115        nationality.set("")
116
117 def UpdateContactWindow(event):
118     global id, UpdateWindow
119     curItem = tree.focus()
120     contents = (tree.item(curItem))
121     selecteditem = contents['values']
122     id = selecteditem[0]
123     f_name.set("")
124     m_name.set("")
125     l_name.set("")
126     gender.set("")
127     age.set("")
128     home_address.set("")
129     phone_number.set("")
130     religion.set("")
131     nationality.set("")
132
133     f_name.set(selecteditem[1])
134     m_name.set(selecteditem[2])

```



```
131 l_name.set(selecteditem[3])
132
133 age.set(selecteditem[5])
134 home_address.set(selecteditem[6])
135 phone_number.set(selecteditem[7])
136 religion.set(selecteditem[8])
137 nationality.set(selecteditem[9])
138
139 UpdateWindow = Toplevel()
140 UpdateWindow.title("Contact Details")
141 UpdateWindow.geometry("500x520+0+0")
142 UpdateWindow.config(bg="dark gray")
143 UpdateWindow.resizable(0, 0)
144 if 'NewWindow' in globals():
145     NewWindow.destroy()
146
147 # FRAMES
148 FormTitle = Frame(UpdateWindow)
149 FormTitle.pack(side=TOP)
150 ContactForm = Frame(UpdateWindow)
151 ContactForm.pack(side=TOP, pady=10)
152 RadioGroup = Frame(ContactForm)
153 Male = Radiobutton(RadioGroup, text="Male", variable=gender, value="Male",
font=('arial', 14)).pack(side=LEFT)
154 Female = Radiobutton(RadioGroup, text="Female", variable=gender,
value="Female", font=('arial', 14)).pack(side=LEFT)
155 Transgender = Radiobutton(RadioGroup, text="Transgender", variable=gender,
value="Transgender", font=('arial', 14)).pack(side=LEFT)
156 # LABELS
157 lbl_title = Label(FormTitle, text="Updating Contacts", bd=12, relief=GROOVE,
fg="White", bg="blue",
158     font=("Calibri", 14, "bold"), pady=3)
159 lbl_title.pack(fill=X)
160 lbl_FirstName = Label(ContactForm, text="First Name", font=('arial', 14), bd=5)
161 lbl_FirstName.grid(row=0, sticky=W)
162
163 lbl_MiddleName = Label(ContactForm, text="Middle Name", font=('arial', 14),
bd=5)
164 lbl_MiddleName.grid(row=1, sticky=W)
165
166 lbl_LastName = Label(ContactForm, text="Last Name", font=('arial', 14), bd=5)
167 lbl_LastName.grid(row=2, sticky=W)
168
169 lbl_Gender = Label(ContactForm, text="Gender", font=('arial', 14), bd=5)
170 lbl_Gender.grid(row=3, sticky=W)
171
172 lbl_Age = Label(ContactForm, text="Age", font=('arial', 14), bd=5)
173 lbl_Age.grid(row=4, sticky=W)
174
```

```
175     lbl_HomeAddress = Label(ContactForm, text=" Home Address", font=('arial', 14),
bd=5)
176     lbl_HomeAddress.grid(row=5, sticky=W)
177
178     lbl_PhoneNumber = Label(ContactForm, text="Phone Number",font=('arial', 14),
bd=5)
179     lbl_PhoneNumber.grid(row=6, sticky=W)
180
181     lbl_Religion = Label(ContactForm, text="Religion", font=('arial', 14), bd=5)
182     lbl_Religion.grid(row=7, sticky=W)
183
184     lbl_Nationality = Label(ContactForm, text="Nationality", font=('arial', 14), bd=5)
185     lbl_Nationality.grid(row=8, sticky=W)
186
187     # TEXT ENTRY
188     FirstName = Entry(ContactForm, textvariable=f_name, font=('arial', 14, 'bold'),
bd=10, width=20, justify='left')
189     FirstName.grid(row=0, column=1)
190
191     MiddleName = Entry(ContactForm, textvariable=m_name, font=('arial', 14, 'bold'),
bd=10, width=20, justify='left')
192     MiddleName.grid(row=1, column=1)
193
194     LastName = Entry(ContactForm, textvariable=l_name, font=('arial', 14, 'bold'),
bd=10, width=20, justify='left')
195     LastName.grid(row=2, column=1)
196
197     RadioGroup.grid(row=3, column=1)
198
199     Age = Entry(ContactForm, textvariable=age, font=('arial', 14, 'bold'), bd=10,
width=20, justify='left')
200     Age.grid(row=4, column=1)
201
202     HomeAddress = Entry(ContactForm, textvariable=home_address, font=('arial', 14,
'bold'), bd=10, width=20,
203         justify='left')
204     HomeAddress.grid(row=5, column=1)
205
206     PhoneNumber = Entry(ContactForm, textvariable=phone_number, font=('arial',
14, 'bold'), bd=10, width=20,
207         justify='left')
208     PhoneNumber.grid(row=6, column=1)
209
210     Religion = Entry(ContactForm, textvariable=religion, font=('arial', 14, 'bold'),
bd=10, width=20, justify='left')
211     Religion.grid(row=7, column=1)
212
213     Nationality = Entry(ContactForm, textvariable=nationality, font=('arial', 14, 'bold'),
bd=10, width=20,
```

```

214             justify='left')
215     Nationality.grid(row=8, column=1)
216     # =====BUTTONS=====
217     ButtonUpdatContact = Button(ContactForm, text='Update', bd=10, font=('arial',
12, 'bold'), relief="ridge", fg="white",
218             bg="blue", command=Update)
219     ButtonUpdatContact.grid(row=9, columnspan=2, pady=10)
220
221
222
223 def Delete():
224     if not tree.selection():
225         result = tkMessageBox.showwarning('', 'Please Select in the Table First!',
icon="warning")
226     else:
227         result = tkMessageBox.askquestion('', 'Are You Sure You Want To Delete This
Record?', icon="warning")
228         if result == 'yes':
229             curItem = tree.focus()
230             contents = (tree.item(curItem))
231             selecteditem = contents['values']
232             tree.delete(curItem)
233             conn = sqlite3.connect("contactdb.db")
234             cursor = conn.cursor()
235             cursor.execute("DELETE FROM `contactable` WHERE `id` = %d" %
selecteditem[0])
236             conn.commit()
237             cursor.close()
238             conn.close()
239
240
241 def AddNewContact():
242     global NewWindow
243     f_name.set("")
244     m_name.set("")
245     l_name.set("")
246     gender.set("")
247     age.set("")
248     home_address.set("")
249     phone_number.set("")
250     religion.set("")
251     nationality.set("")
252     NewWindow = Toplevel()
253     NewWindow.title("Contact Details")
254     NewWindow.resizable(0, 0)
255     NewWindow.geometry("500x520+0+0")
256     NewWindow.config(bg="dark gray")
257     if 'UpdateWindow' in globals():
258         UpdateWindow.destroy()

```

```

259
260 # =====FRAMES=====
261 FormTitle = Frame(NewWindow)
262 FormTitle.pack(side=TOP)
263 ContactForm = Frame(NewWindow)
264 ContactForm.pack(side=TOP, pady=10)
265 RadioGroup = Frame(ContactForm)
266 Male = Radiobutton(RadioGroup, text="Male", variable=gender, value="Male",
font=('arial', 14)).pack(side=LEFT)
267 Female = Radiobutton(RadioGroup, text="Female", variable=gender,
value="Female", font=('arial', 14)).pack(side=LEFT)
268 Transgender = Radiobutton(RadioGroup, text="Transgender", variable=gender,
value="Transgender", font=('arial', 14)).pack(side=LEFT)
269 # =====LABELS=====
270 lbl_title = Label(FormTitle, text="Adding New Contacts", bd=12,
relief=GROOVE, fg="White", bg="blue",
271 font=('Calibri', 14, "bold"), pady=3)
272 lbl_title.pack(fill=X)
273 lbl_FirstName = Label(ContactForm, text="First Name", font=('arial', 14), bd=5)
274 lbl_FirstName.grid(row=0, sticky=W)
275
276 lbl_MiddleName = Label(ContactForm, text="Middle Name", font=('arial', 14),
bd=5)
277 lbl_MiddleName.grid(row=1, sticky=W)
278
279 lbl_LastName = Label(ContactForm, text="Last Name", font=('arial', 14), bd=5)
280 lbl_LastName.grid(row=2, sticky=W)
281
282 lbl_Gender = Label(ContactForm, text="Gender", font=('arial', 14), bd=5)
283 lbl_Gender.grid(row=3, sticky=W)
284
285 lbl_Age = Label(ContactForm, text="Age", font=('arial', 14), bd=5)
286 lbl_Age.grid(row=4, sticky=W)
287
288 lbl_HomeAddress = Label(ContactForm, text=" Home Address", font=('arial', 14),
bd=5)
289 lbl_HomeAddress.grid(row=5, sticky=W)
290
291 lbl_PhoneNumber = Label(ContactForm, text="Phone Number", font=('arial', 14),
bd=5)
292 lbl_PhoneNumber.grid(row=6, sticky=W)
293
294 lbl_Religion = Label(ContactForm, text="Religion", font=('arial', 14), bd=5)
295 lbl_Religion.grid(row=7, sticky=W)
296
297 lbl_Nationality = Label(ContactForm, text="Nationality", font=('arial', 14), bd=5)
298 lbl_Nationality.grid(row=8, sticky=W)
299
300 # =====ENTRY=====

```

```

301  FirstName = Entry(ContactForm, textvariable=f_name, font=('arial', 14, 'bold'),
bd=10, width=20, justify='left')
302  FirstName.grid(row=0, column=1)
303
304  MiddleName = Entry(ContactForm, textvariable=m_name, font=('arial', 14, 'bold'),
bd=10, width=20, justify='left')
305  MiddleName.grid(row=1, column=1)
306
307  LastName = Entry(ContactForm, textvariable=l_name, font=('arial', 14, 'bold'),
bd=10, width=20, justify='left')
308  LastName.grid(row=2, column=1)
309
310  RadioGroup.grid(row=3, column=1)
311
312  Age = Entry(ContactForm, textvariable=age, font=('arial', 14, 'bold'), bd=10,
width=20, justify='left')
313  Age.grid(row=4, column=1)
314
315  HomeAddress = Entry(ContactForm, textvariable=home_address, font=('arial', 14,
'bold'), bd=10, width=20, justify='left')
316  HomeAddress.grid(row=5, column=1)
317
318  PhoneNumber = Entry(ContactForm, textvariable=phone_number, font=('arial',
14, 'bold'), bd=10, width=20, justify='left')
319  PhoneNumber.grid(row=6, column=1)
320
321  Religion = Entry(ContactForm, textvariable=religion, font=('arial', 14, 'bold'),
bd=10, width=20, justify='left')
322  Religion.grid(row=7, column=1)
323
324  Nationality = Entry(ContactForm, textvariable=nationality, font=('arial', 14, 'bold'),
bd=10, width=20, justify='left')
325  Nationality.grid(row=8, column=1)
326
327  # =====BUTTONS=====
328  ButtonAddContact = Button(ContactForm, text='Save', bd=10, font=('arial', 12,
'bold'), relief="ridge", fg="white",
329      bg="light blue", command=Submit)
330  ButtonAddContact.grid(row=9, columnspan=2, pady=10)
331
332
333  #
=====FRAMES=====
=====
334  Top = Frame(root, width=500, bd=1, relief=SOLID)
335  Top.pack(side=TOP)
336  Mid = Frame(root, width=500, bg="light blue")
337  Mid.pack(side=BOTTOM)
338  f1 = Frame(root, width=6, height=8, bd=8, bg="light blue")

```

```

339 fl.pack(side=BOTTOM)
340 flb = Frame(fl, width=6, height=8, bd=8, bg="light blue")
341 flb.pack(side=BOTTOM)
342 MidLeft = Frame(Mid, width=100)
343 MidLeft.pack(side=LEFT, pady=10)
344 MidLeftPadding = Frame(Mid, width=370, bg="light blue")
345 MidLeftPadding.pack(side=LEFT)
346 MidRight = Frame(Mid, width=100)
347 MidRight.pack(side=RIGHT, pady=10)
348 TableMargin = Frame(root, width=500)
349 TableMargin.pack(side=TOP)
350
351 # LABELS
352 lbl_title = Label(Top, text="Contact Book", bd=12, relief=GROOVE, fg="black",
353                    bg="peachpuff2",
354                    font=("Times", 36, "bold italic  "), pady=3)
355 lbl_title.pack(fill=X)
356
357 # BUTTONS
358 ButtonAdd = Button(fl, text='Add New Contact', bd=8, font=('arial', 12, 'bold'),
359                    relief="groove", fg="black",
360                    bg="dark gray", command=AddNewContact).grid(row=0, column=0,
361                    padx=20, pady=30)
362
363 ButtonDelete = Button(fl, text='Delete', bd=8, font=('arial', 12, 'bold'),
364                       relief="groove", command=Delete,
365                       fg="black", bg="dark gray").grid(row=0, column=1, padx=20)
366
367 ButtonExit = Button(fl, text='Exit System', bd=8, font=('arial', 12, 'bold'),
368                     relief="groove", command=Exit,
369                     fg="black", bg="dark gray").grid(row=0, column=2, padx=20, pady=30)
370
371 # TABLES
372 scrollbarx = Scrollbar(TableMargin, orient=HORIZONTAL)
373 scrollbary = Scrollbar(TableMargin, orient=VERTICAL)
374 tree = ttk.Treeview(TableMargin, columns=("Id", "First Name", "Middle Name",
375 "Last Name", "Gender", "Age", "Home Address", "Phone Number", "Religion",
376 "Nationality"),
377                    height=400, selectmode="extended", yscrollcommand=scrollbary.set,
378                    xscrollcommand=scrollbarx.set)
379 scrollbary.config(command=tree.yview)
380 scrollbarx.pack(side=RIGHT, fill=Y)
381 scrollbary.pack(side=LEFT, fill=Y)
382 scrollbarx.config(command=tree.xview)
383 scrollbarx.pack(side=BOTTOM, fill=X)
384 tree.heading('Id', text="Id", anchor=W)
385 tree.heading('First Name', text="First Name", anchor=W)
386 tree.heading('Middle Name', text="Middle Name", anchor=W)
387 tree.heading('Last Name', text="Last Name", anchor=W)

```

```

380 tree.heading('Gender', text="Gender", anchor=W)
381 tree.heading('Age', text="Age", anchor=W)
382 tree.heading('Home Address', text="Home Address", anchor=W)
383 tree.heading('Phone Number', text="phone Number", anchor=W)
384 tree.heading('Religion', text="Religion", anchor=W)
385 tree.heading('Nationality', text="Nationality", anchor=W)
386 tree.column('#0', stretch=NO, minwidth=0, width=0)
387 tree.column('#1', stretch=NO, minwidth=0, width=0)
388 tree.column('#2', stretch=NO, minwidth=0, width=80)
389 tree.column('#3', stretch=NO, minwidth=0, width=120)
390 tree.column('#4', stretch=NO, minwidth=0, width=90)
391 tree.column('#5', stretch=NO, minwidth=0, width=80)
392 tree.column('#6', stretch=NO, minwidth=0, width=30)
393 tree.column('#7', stretch=NO, minwidth=0, width=120)
394 tree.column('#8', stretch=NO, minwidth=0, width=120)
395 tree.column('#9', stretch=NO, minwidth=0, width=120)
396 tree.pack()
397 tree.bind('<Double-Button-1>', UpdateContactWindow)
398
399 #
=====INITIALIZATION=====
=====
400 if __name__ == '__main__':
401     Database()
402     root.mainloop()

```

Results

We finally got the end product as a '*Conatct Management System*' that includes all the mentioned modules. We learnt how to make a GUI using Tkinter in Python and also learnt to implement database connectivity using sqlite3.

This project aims at Business Process automation by providing a user friendly interface to the users to let them easily use the contact book . Through this project we aim to target the easy to understand by the user and operator and good user interface.

>Easy to use .

>Be easy to operate.

>Manages the information of profiles.

>shows the information about the contact.

>To increase efficiency of managing the contact.

>Edit ,Add, update of records to improved which result in proper resource management of contact data.

>work becomes very speedy.

References

- <https://www.python-course.eu/index.php>
- www.blog.pythonlibrary.org
- <http://effbot.org/tkinterbook/>
- <http://www.dealingdata.net/2016/08/21/Python-MySQL-GUI/>
- <https://python-forum.io/index.php>
- <https://www.w3schools.com/python/>
- <https://www.geeksforgeeks.org/sql-using-python/>
- <https://stackoverflow.com/>
- www.tutorialspoint.com
- www.reddit.com
- www.google.co.in
- www.quora.com