CONTACT MANAGEMENT SYSTEM

END-TERM REPORT

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By:

S.no.	*Name*	*Roll No.*	*Registration no.*
1.	*AKHIL NEELAM*	*01*	*12001936*

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**

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

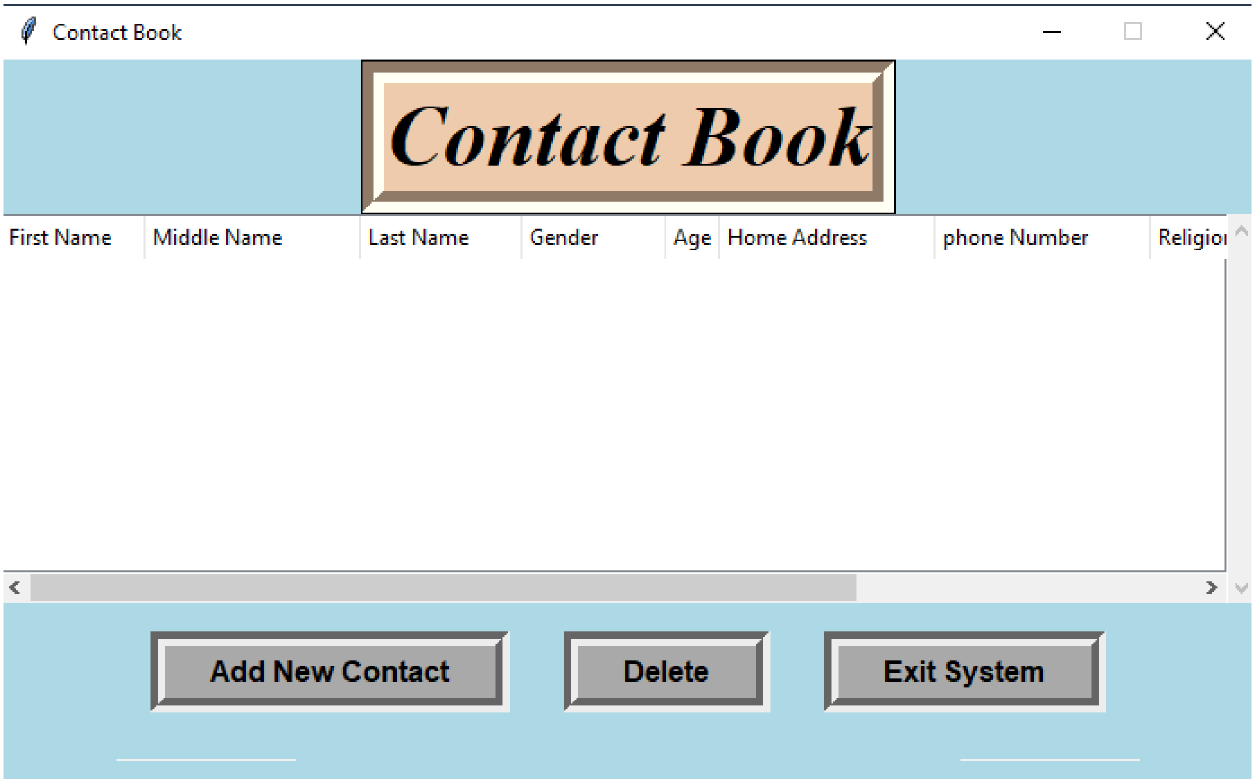
Delete: This Module is to Delete the unwanted contact from database

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

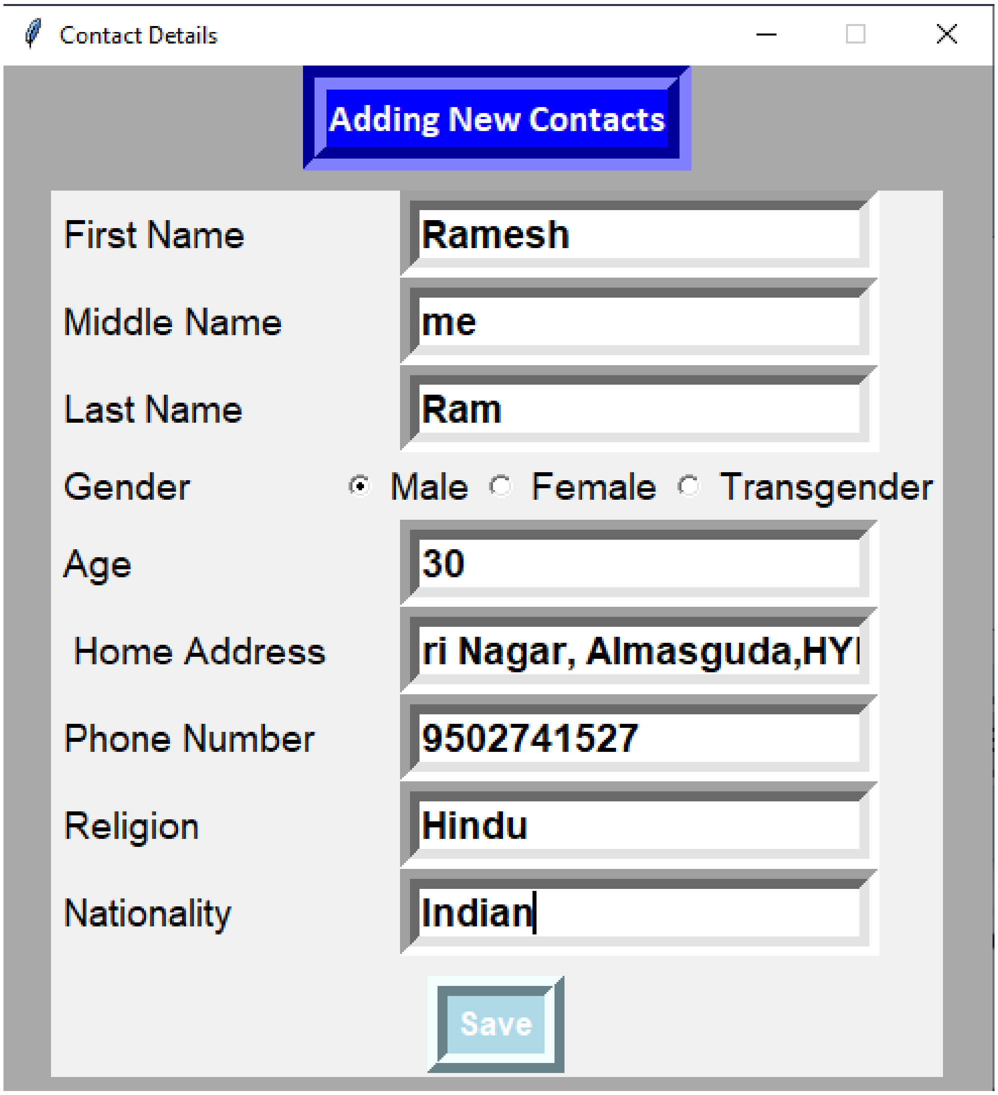
Exit the system: This Module is to exit from the contact management System

GUI Screenshots:

1.WELCOME SCREEN:

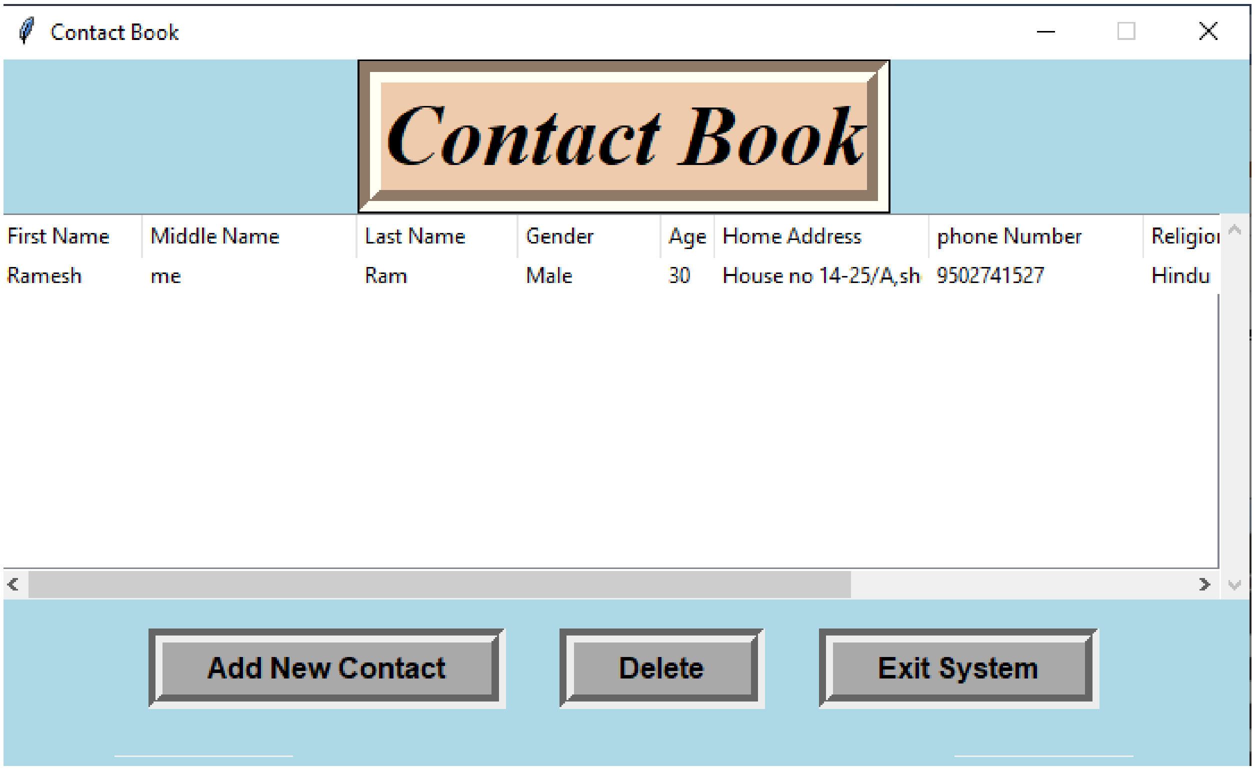


2.Add New Contact (*with input values*)



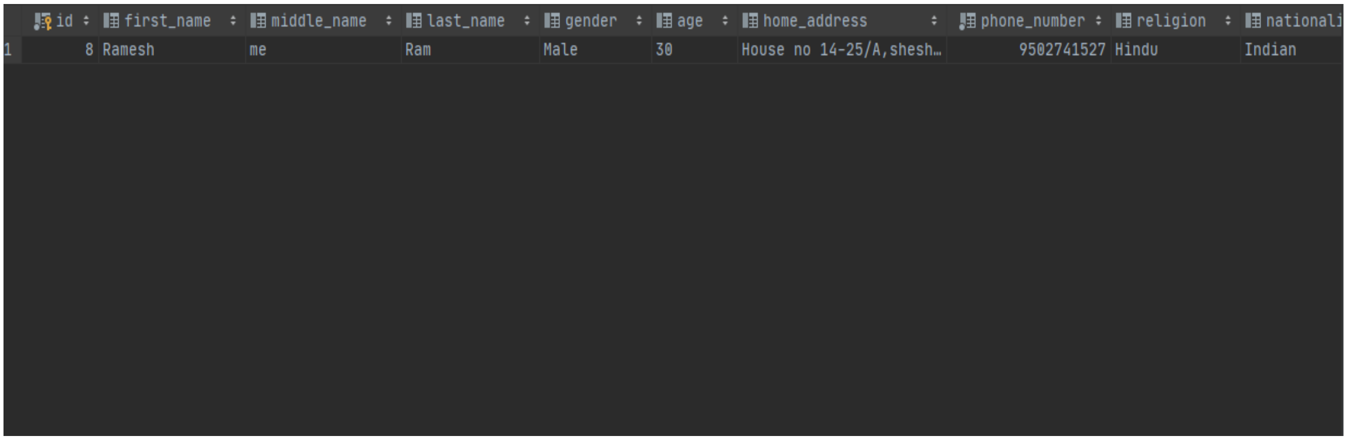
First Name	Ramesh
Middle Name	me
Last Name	Ram
Gender	● Male ○ Female ○ Transgender
Age	30
Home Address	ri Nagar, Almasguda,HYI
Phone Number	9502741527
Religion	Hindu
Nationality	Indian

3. Save of the Contact (*with input values*)



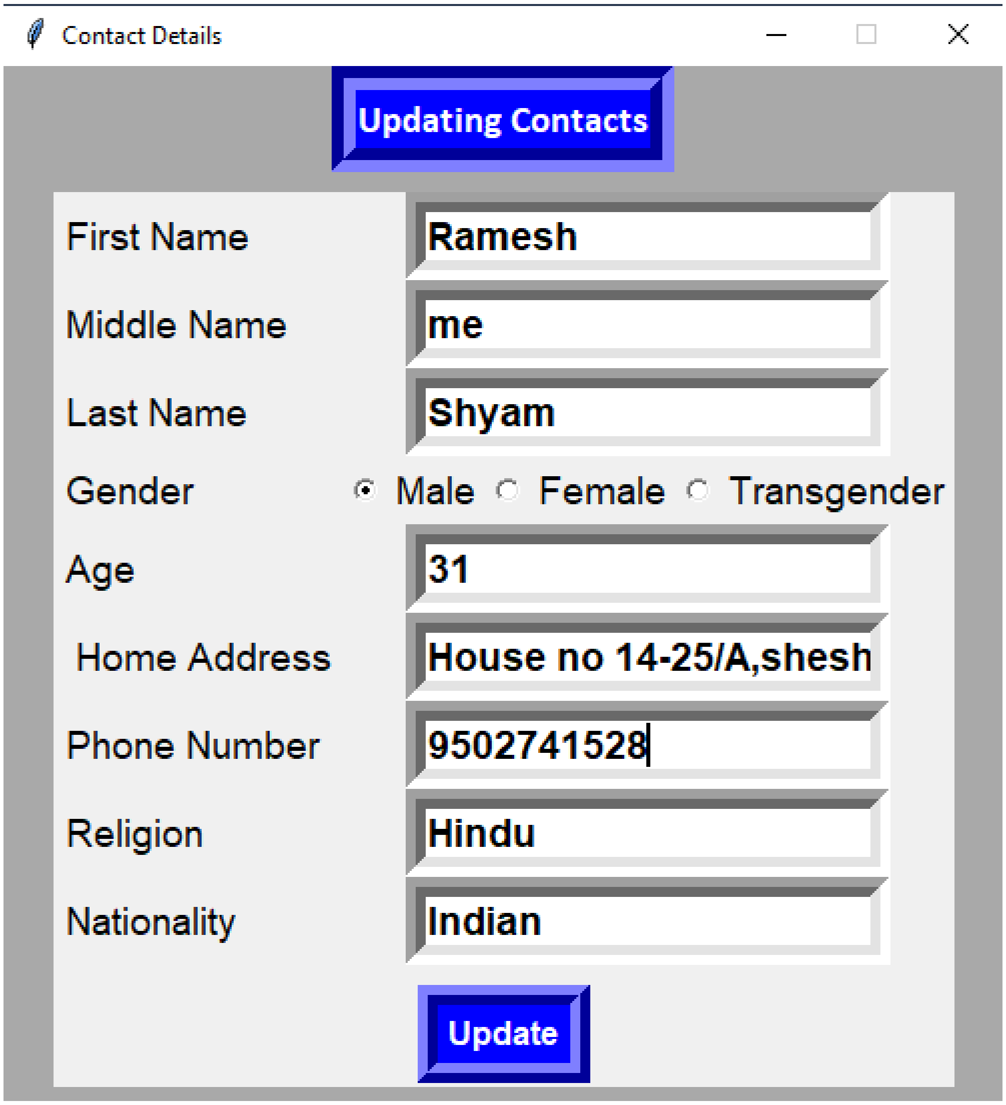
First Name	Middle Name	Last Name	Gender	Age	Home Address	phone Number	Religioı
Ramesh	me	Ram	Male	30	House no 14-25/A,sh	9502741527	Hindu

4. Save in the database:



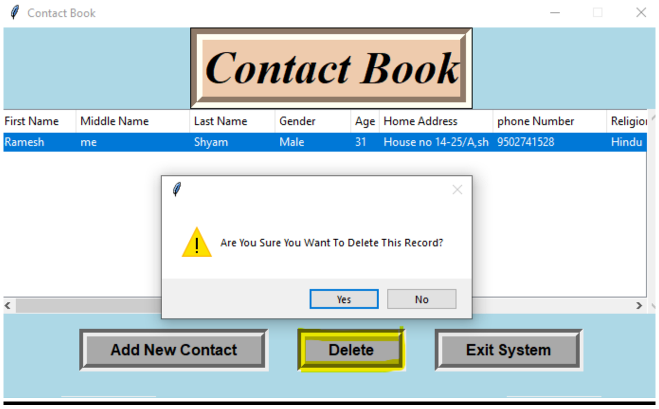
	id	first_name	middle_name	last_name	gender	age	home_address	phone_number	religion	nationali
1	8	Ramesh	me	Ram	Male	30	House no 14-25/A,shesh…	9502741527	Hindu	Indian

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



First Name	Ramesh
Middle Name	me
Last Name	Shyam
Gender	● Male ○ Female ○ Transgender
Age	31
Home Address	House no 14-25/A,shesh
Phone Number	9502741528
Religion	Hindu
Nationality	Indian

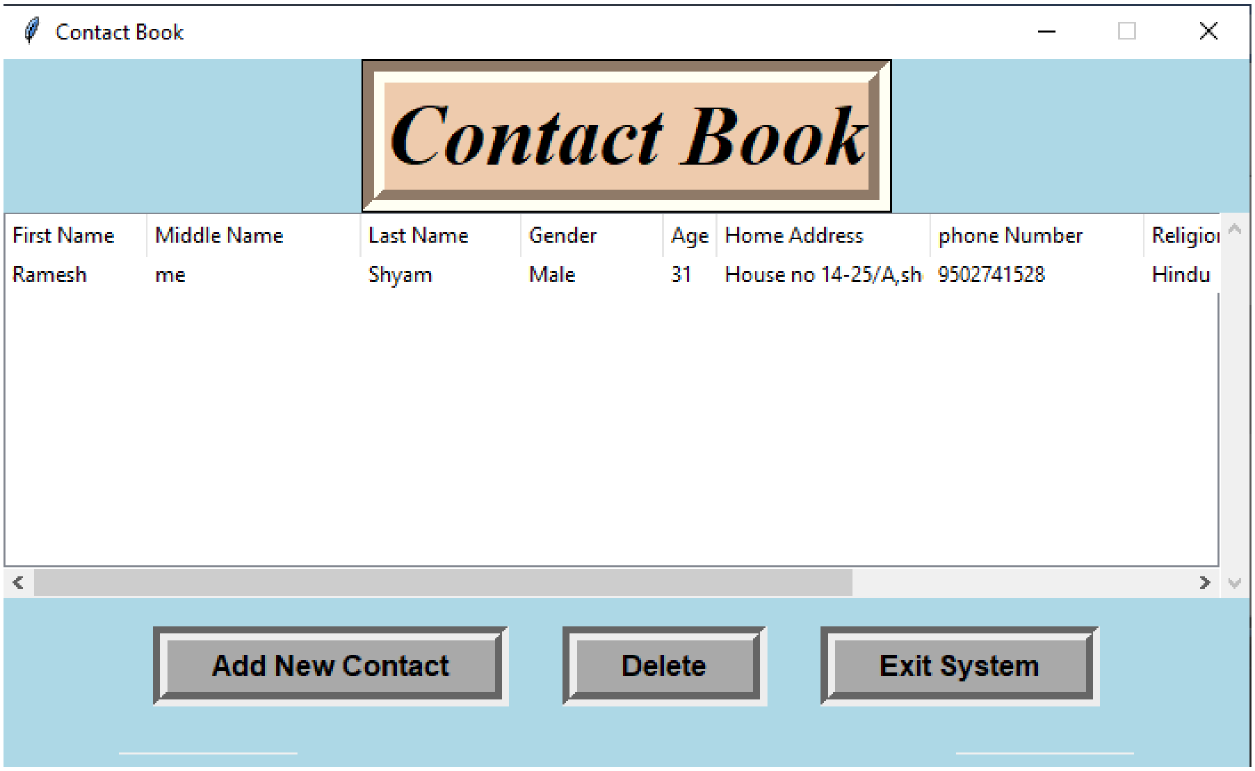
Delete the existing contact: (*on clicking the contact which want to delete click yes to delete the contact)*



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

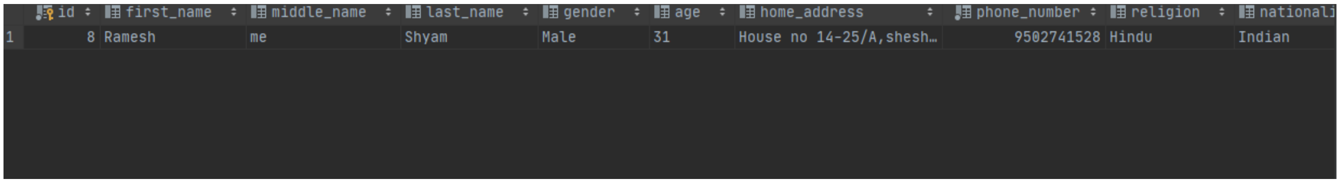
Are You Sure You Want To Delete This Record?

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



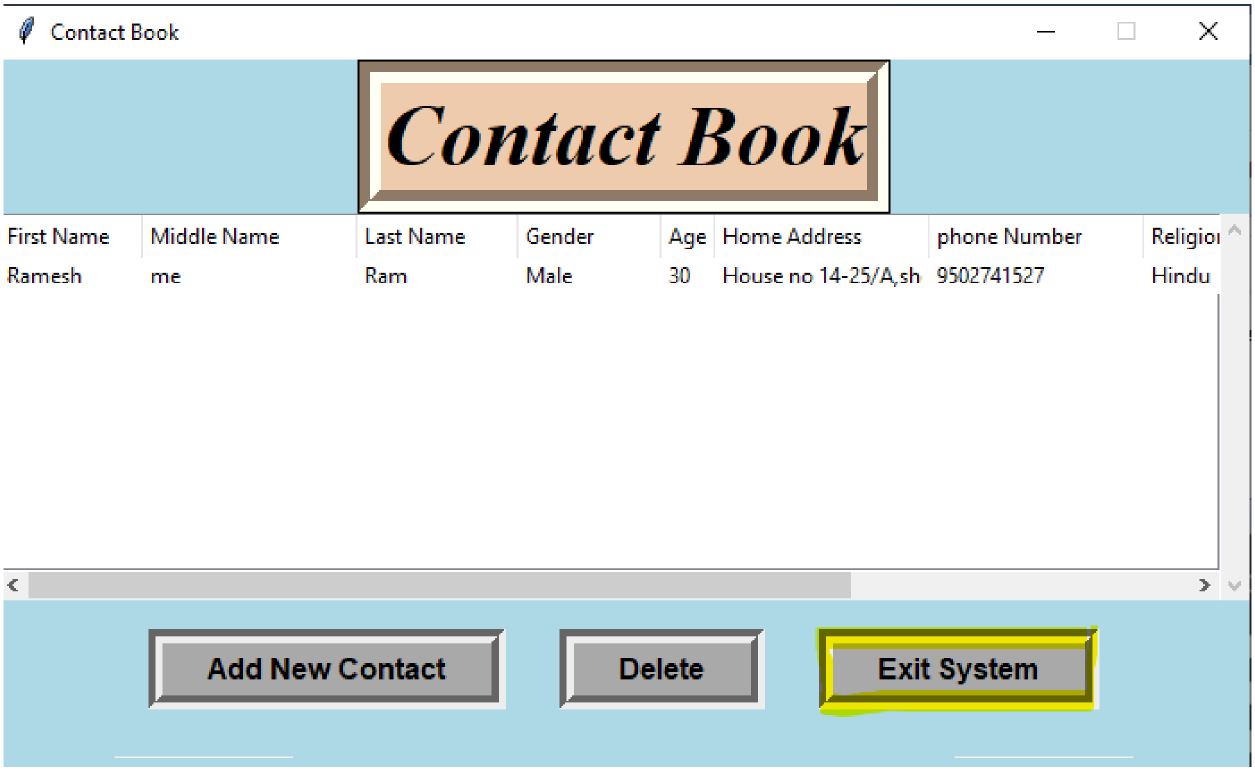
First Name	Middle Name	Last Name	Gender	Age	Home Address	phone Number	Religioı
Ramesh	me	Shyam	Male	31	House no 14-25/A,sh	9502741528	Hindu

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	nationali
1	8	Ramesh	me	Shyam	Male	31	House no 14-25/A,shesh…	9502741528	Hindu	Indian

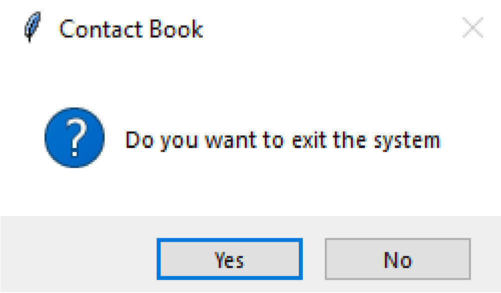
8. Exit System (*on clicking Exit system*)



Contact Book

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

(*Click Yes to exit from the system)*



Contact Book

Do you want to exit the system

Source Code

```
from tkinter import *
import sqlite3
import tkinter
import tkinter.ttk as ttk
import tkinter.messagebox as tkMessageBox


root = Tk()
root.title("Contact Book")
root.geometry("700x400+0+0")
root.resizable(0, 0)
root.config(bg="light blue")

# VARIABLES
f_name = StringVar()
m_name = StringVar()
l_name = StringVar()
age = StringVar()
home_address = StringVar()
phone_number = StringVar()
gender = StringVar()
religion = StringVar()
nationality = StringVar()

# METHODS
def Exit():
    wayOut = tkinter.messagebox.askyesno("Contact Book", "Do you want to exit the system")
    if wayOut > 0:
        root.destroy()
        return

def Reset():
    f_name.set("")
    m_name.set("")
    l_name.set("")
    gender.set("")
    age.set("")
    home_address.set("")
    phone_number.set("")
    religion.set("")
    nationality.set("")

def Database():
    conn = sqlite3.connect("contactdb.db")
    cursor = conn.cursor()
    cursor.execute(
        "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)")
    cursor.execute("SELECT * FROM `contactable` ORDER BY `last_name` ASC")
    fetch = cursor.fetchall()
    for data in fetch:
        tree.insert('', 'end', values=(data))
    cursor.close()
    conn.close()


def Submit():
    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:
        result = tkMessageBox.showwarning('', 'Please Complete The Required Field', icon="warning")
    else:
        tree.delete(*tree.get_children())
        conn = sqlite3.connect("contactdb.db")
        cursor = conn.cursor()
        cursor.execute(
            "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()),
            int(phone_number.get()), str(religion.get()), str(nationality.get())))
        conn.commit()
        cursor.execute("SELECT * FROM `contactable` ORDER BY `last_name` ASC")
        fetch = cursor.fetchall()
        for data in fetch:
            tree.insert('', 'end', values=(data))
        cursor.close()
        conn.close()
        f_name.set("")
        m_name.set("")
        l_name.set("")
        gender.set("")
        age.set("")
        home_address.set("")
        phone_number.set("")
        religion.set("")
        nationality.set("")


def Update():
    if gender.get() == "" or phone_number.get() == str:
        result = tkMessageBox.showwarning('', 'Please Complete The Required Field ', icon="warning")
    else:
        tree.delete(*tree.get_children())
        conn = sqlite3.connect("contactdb.db")
        cursor = conn.cursor()
        cursor.execute(
            "UPDATE `contactable` SET `first_name` = ?, `middle_name` = ? , `last_name` = ?, `gender` =?, `age` = ?,  `home_address` = ?, `phone_number` = ?, `religion` = ?, `nationality` = ? WHERE `id` = ?",
            (str(f_name.get()), str(m_name.get()),  str(l_name.get()), str(gender.get()), int(age.get()), str(home_address.get()),
            int(phone_number.get()), str(religion.get()), str(nationality.get()), int(id)))
        conn.commit()
        cursor.execute("SELECT * FROM `contactable` ORDER BY `last_name` ASC")
        fetch = cursor.fetchall()
        for data in fetch:
            tree.insert('', 'end', values=(data))
        cursor.close()
        conn.close()
        f_name.set("")
        m_name.set("")
        l_name.set("")
        gender.set("")
        age.set("")
        home_address.set("")
        phone_number.set("")
        religion.set("")
        nationality.set("")

def UpdateContactWindow(event):
    global id, UpdateWindow
    curItem = tree.focus()
    contents = (tree.item(curItem))
    selecteditem = contents['values']
    id = selecteditem[0]
    f_name.set("")
    m_name.set("")
    l_name.set("")
    gender.set("")
    age.set("")
    home_address.set("")
    phone_number.set("")
    religion.set("")
    nationality.set("")

    f_name.set(selecteditem[1])
    m_name.set(selecteditem[2])
```

```
l_name.set(selecteditem[3])

age.set(selecteditem[5])
home_address.set(selecteditem[6])
phone_number.set(selecteditem[7])
religion.set(selecteditem[8])
nationality.set(selecteditem[9])

UpdateWindow = Toplevel()
UpdateWindow.title("Contact Details")
UpdateWindow.geometry("500x520+0+0")
UpdateWindow.config(bg="dark gray")
UpdateWindow.resizable(0, 0)
if 'NewWindow' in globals():
    NewWindow.destroy()

# FRAMES
FormTitle = Frame(UpdateWindow)
FormTitle.pack(side=TOP)
ContactForm = Frame(UpdateWindow)
ContactForm.pack(side=TOP, pady=10)
RadioGroup = Frame(ContactForm)
Male = Radiobutton(RadioGroup, text="Male", variable=gender, value="Male", font=('arial', 14)).pack(side=LEFT)
Female = Radiobutton(RadioGroup, text="Female", variable=gender, value="Female", font=('arial', 14)).pack(side=LEFT)
Transgender = Radiobutton(RadioGroup, text="Transgender", variable=gender, value="Transgender", font=('arial', 14)).pack(side=LEFT)
# LABELS
lbl_title = Label(FormTitle, text="Updating Contacts", bd=12, relief=GROOVE, fg="White", bg="blue",
                  font=("Calibri", 14, "bold"), pady=3)
lbl_title.pack(fill=X)
lbl_FirstName = Label(ContactForm, text="First Name", font=('arial', 14), bd=5)
lbl_FirstName.grid(row=0, sticky=W)

lbl_MiddleName = Label(ContactForm, text="Middle Name", font=('arial', 14), bd=5)
lbl_MiddleName.grid(row=1, sticky=W)

lbl_LastName = Label(ContactForm, text="Last Name", font=('arial', 14), bd=5)
lbl_LastName.grid(row=2, sticky=W)

lbl_Gender = Label(ContactForm, text="Gender", font=('arial', 14), bd=5)
lbl_Gender.grid(row=3, sticky=W)

lbl_Age = Label(ContactForm, text="Age", font=('arial', 14), bd=5)
lbl_Age.grid(row=4, sticky=W)

```

```
lbl_HomeAddress = Label(ContactForm, text=" Home Address", font=('arial', 14), bd=5)
lbl_HomeAddress.grid(row=5, sticky=W)

lbl_PhoneNumber = Label(ContactForm, text="Phone Number",font=('arial', 14), bd=5)
lbl_PhoneNumber.grid(row=6, sticky=W)

lbl_Religion = Label(ContactForm, text="Religion", font=('arial', 14), bd=5)
lbl_Religion.grid(row=7, sticky=W)

lbl_Nationality = Label(ContactForm, text="Nationality", font=('arial', 14), bd=5)
lbl_Nationality.grid(row=8, sticky=W)

# TEXT ENTRY
FirstName = Entry(ContactForm, textvariable=f_name, font=('arial', 14, 'bold'), bd=10, width=20, justify='left')
FirstName.grid(row=0, column=1)

MiddleName = Entry(ContactForm, textvariable=m_name, font=('arial', 14, 'bold'), bd=10, width=20, justify='left')
MiddleName.grid(row=1, column=1)

LastName = Entry(ContactForm, textvariable=l_name, font=('arial', 14, 'bold'), bd=10, width=20, justify='left')
LastName.grid(row=2, column=1)

RadioGroup.grid(row=3, column=1)

Age = Entry(ContactForm, textvariable=age, font=('arial', 14, 'bold'), bd=10, width=20, justify='left')
Age.grid(row=4, column=1)

HomeAddress = Entry(ContactForm, textvariable=home_address, font=('arial', 14, 'bold'), bd=10, width=20,
                    justify='left')
HomeAddress.grid(row=5, column=1)

PhoneNumber = Entry(ContactForm, textvariable=phone_number, font=('arial', 14, 'bold'), bd=10, width=20,
                    justify='left')
PhoneNumber.grid(row=6, column=1)

Religion = Entry(ContactForm, textvariable=religion, font=('arial', 14, 'bold'), bd=10, width=20, justify='left')
Religion.grid(row=7, column=1)

Nationality = Entry(ContactForm, textvariable=nationality, font=('arial', 14, 'bold'), bd=10, width=20,
                     justify='left')
    Nationality.grid(row=8, column=1)
    # ==================BUTTONS==============================
    ButtonUpdatContact = Button(ContactForm, text='Update', bd=10, font=('arial', 12, 'bold'), relief="ridge", fg="white",
                            bg="blue", command=Update)
    ButtonUpdatContact.grid(row=9, columnspan=2, pady=10)



def Delete():
    if not tree.selection():
        result = tkMessageBox.showwarning('', 'Please Select in the Table First!', icon="warning")
    else:
        result = tkMessageBox.askquestion('', 'Are You Sure You Want To Delete This Record?', icon="warning")
        if result == 'yes':
            curItem = tree.focus()
            contents = (tree.item(curItem))
            selecteditem = contents['values']
            tree.delete(curItem)
            conn = sqlite3.connect("contactdb.db")
            cursor = conn.cursor()
            cursor.execute("DELETE FROM `contactable` WHERE `id` = %d" % selecteditem[0])
            conn.commit()
            cursor.close()
            conn.close()


def AddNewContact():
    global NewWindow
    f_name.set("")
    m_name.set("")
    l_name.set("")
    gender.set("")
    age.set("")
    home_address.set("")
    phone_number.set("")
    religion.set("")
    nationality.set("")
    NewWindow = Toplevel()
    NewWindow.title("Contact Details")
    NewWindow.resizable(0, 0)
    NewWindow.geometry("500x520+0+0")
    NewWindow.config(bg="dark gray")
    if 'UpdateWindow' in globals():
        UpdateWindow.destroy()
```

```

# ===================FRAMES================================
FormTitle = Frame(NewWindow)
FormTitle.pack(side=TOP)
ContactForm = Frame(NewWindow)
ContactForm.pack(side=TOP, pady=10)
RadioGroup = Frame(ContactForm)
Male = Radiobutton(RadioGroup, text="Male", variable=gender, value="Male", font=('arial', 14)).pack(side=LEFT)
Female = Radiobutton(RadioGroup, text="Female", variable=gender, value="Female", font=('arial', 14)).pack(side=LEFT)
Transgender = Radiobutton(RadioGroup, text="Transgender", variable=gender, value="Transgender", font=('arial', 14)).pack(side=LEFT)
# ===================LABELS================================
lbl_title = Label(FormTitle, text="Adding New Contacts",  bd=12, relief=GROOVE, fg="White", bg="blue",
                    font=("Calibri", 14, "bold"), pady=3)
lbl_title.pack(fill=X)
lbl_FirstName = Label(ContactForm, text="First Name", font=('arial', 14), bd=5)
lbl_FirstName.grid(row=0, sticky=W)

lbl_MiddleName = Label(ContactForm, text="Middle Name", font=('arial', 14), bd=5)
lbl_MiddleName.grid(row=1, sticky=W)

lbl_LastName = Label(ContactForm, text="Last Name", font=('arial', 14), bd=5)
lbl_LastName.grid(row=2, sticky=W)

lbl_Gender = Label(ContactForm, text="Gender", font=('arial', 14), bd=5)
lbl_Gender.grid(row=3, sticky=W)

lbl_Age = Label(ContactForm, text="Age", font=('arial', 14), bd=5)
lbl_Age.grid(row=4, sticky=W)

lbl_HomeAddress = Label(ContactForm, text=" Home Address", font=('arial', 14), bd=5)
lbl_HomeAddress.grid(row=5, sticky=W)

lbl_PhoneNumber = Label(ContactForm, text="Phone Number", font=('arial', 14), bd=5)
lbl_PhoneNumber.grid(row=6, sticky=W)

lbl_Religion = Label(ContactForm, text="Religion", font=('arial', 14), bd=5)
lbl_Religion.grid(row=7, sticky=W)

lbl_Nationality = Label(ContactForm, text="Nationality", font=('arial', 14), bd=5)
lbl_Nationality.grid(row=8, sticky=W)

# ===================ENTRY================================
    FirstName = Entry(ContactForm, textvariable=f_name, font=('arial', 14, 'bold'), bd=10, width=20, justify='left')
    FirstName.grid(row=0, column=1)

    MiddleName = Entry(ContactForm, textvariable=m_name, font=('arial', 14, 'bold'), bd=10, width=20, justify='left')
    MiddleName.grid(row=1, column=1)

    LastName = Entry(ContactForm, textvariable=l_name, font=('arial', 14, 'bold'), bd=10, width=20, justify='left')
    LastName.grid(row=2, column=1)

    RadioGroup.grid(row=3, column=1)

    Age = Entry(ContactForm, textvariable=age, font=('arial', 14, 'bold'), bd=10, width=20, justify='left')
    Age.grid(row=4, column=1)

    HomeAddress = Entry(ContactForm, textvariable=home_address, font=('arial', 14, 'bold'), bd=10, width=20, justify='left')
    HomeAddress.grid(row=5, column=1)

    PhoneNumber = Entry(ContactForm, textvariable=phone_number, font=('arial', 14, 'bold'), bd=10, width=20, justify='left')
    PhoneNumber.grid(row=6, column=1)

    Religion = Entry(ContactForm, textvariable=religion, font=('arial', 14, 'bold'), bd=10, width=20, justify='left')
    Religion.grid(row=7, column=1)

    Nationality = Entry(ContactForm, textvariable=nationality, font=('arial', 14, 'bold'), bd=10, width=20, justify='left')
    Nationality.grid(row=8, column=1)

    # ==================BUTTONS==============================
    ButtonAddContact = Button(ContactForm, text='Save',  bd=10, font=('arial', 12, 'bold'), relief="ridge", fg="white",
                              bg="light blue", command=Submit)
    ButtonAddContact.grid(row=9, columnspan=2, pady=10)


# ============================FRAMES======================================
Top = Frame(root, width=500, bd=1, relief=SOLID)
Top.pack(side=TOP)
Mid = Frame(root, width=500, bg="light blue")
Mid.pack(side=BOTTOM)
f1 = Frame(root, width=6, height=8, bd=8, bg="light blue")
```

```
f1.pack(side=BOTTOM)
flb = Frame(f1, width=6, height=8, bd=8, bg="light blue")
flb.pack(side=BOTTOM)
MidLeft = Frame(Mid, width=100)
MidLeft.pack(side=LEFT, pady=10)
MidLeftPadding = Frame(Mid, width=370, bg="light blue")
MidLeftPadding.pack(side=LEFT)
MidRight = Frame(Mid, width=100)
MidRight.pack(side=RIGHT, pady=10)
TableMargin = Frame(root, width=500)
TableMargin.pack(side=TOP)

# LABELS
lbl_title = Label(Top,  text="Contact Book", bd=12, relief=GROOVE, fg="black", bg="peachpuff2",
                  font=("Times", 36, "bold italic   "), pady=3)
lbl_title.pack(fill=X)


# BUTTONS
ButtonAdd = Button(flb, text='Add New Contact',  bd=8, font=('arial', 12, 'bold'), relief="groove", fg="black",
                 bg="dark gray", command=AddNewContact).grid(row=0, column=0, ipadx=20, padx=30)

ButtonDelete = Button(flb, text='Delete', bd=8, font=('arial', 12, 'bold'), relief="groove",  command=Delete,
                 fg="black", bg="dark gray").grid(row=0, column=1, ipadx=20)

ButtonExit = Button(flb, text='Exit System', bd=8, font=('arial', 12, 'bold'), relief="groove",  command=Exit,
                fg="black", bg="dark gray").grid(row=0, column=2, ipadx=20, padx=30)

# TABLES
scrollbarx = Scrollbar(TableMargin, orient=HORIZONTAL)
scrollbary = Scrollbar(TableMargin, orient=VERTICAL)
tree = ttk.Treeview(TableMargin, columns=("Id", "First Name", "Middle Name", "Last Name", "Gender", "Age", "Home Address", "Phone Number", "Religion", "Nationality"),
                 height=400, selectmode="extended", yscrollcommand=scrollbary.set, xscrollcommand=scrollbarx.set)
scrollbary.config(command=tree.yview)
scrollbary.pack(side=RIGHT, fill=Y)
scrollbarx.config(command=tree.xview)
scrollbarx.pack(side=BOTTOM, fill=X)
tree.heading('Id', text="Id", anchor=W)
tree.heading('First Name', text="First Name", anchor=W)
tree.heading('Middle Name', text="Middle Name", anchor=W)
tree.heading('Last Name', text="Last Name", anchor=W)
tree.heading('Gender', text="Gender", anchor=W)
tree.heading('Age', text="Age", anchor=W)
tree.heading('Home Address', text="Home Address", anchor=W)
tree.heading('Phone Number', text="phone Number", anchor=W)
tree.heading('Religion', text="Religion", anchor=W)
tree.heading('Nationality', text="Nationality", anchor=W)
tree.column('#0', stretch=NO, minwidth=0, width=0)
tree.column('#1', stretch=NO, minwidth=0, width=0)
tree.column('#2', stretch=NO, minwidth=0, width=80)
tree.column('#3', stretch=NO, minwidth=0, width=120)
tree.column('#4', stretch=NO, minwidth=0, width=90)
tree.column('#5', stretch=NO, minwidth=0, width=80)
tree.column('#6', stretch=NO, minwidth=0, width=30)
tree.column('#7', stretch=NO, minwidth=0, width=120)
tree.column('#8', stretch=NO, minwidth=0, width=120)
tree.column('#9', stretch=NO, minwidth=0, width=120)
tree.pack()
tree.bind('<Double-Button-1>', UpdateContactWindow)

#
=============================INITIALIZATION==========================
=====
if __name__ == '__main__':
    Database()
    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