PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

100 Feet Ring Road, BSK III Stage, Bengaluru-560 085

Department of Computer Science & Engineering

Topic: Phonebook

Team Members

PES1UG21CS574 – Shreya Mishra

PES1UG21CS603 - Skanda Shreesha Prashad

PES1UG21CS598 - Isham Sinha

ABSTRACT

A phone book is a collection of the data required to contact certain individuals such as their name, phone numbers, address, email, etc. We have created a virtual phone book using Python. The user has to input the data in the various fields present and the project converts them into a list. Graphical User Interface has been added using Tkinter and various concepts such as importing functions from another file, timestamp, etc., have been used. The project was made using PyCharm and Visual Studio Code.

TABLE OF CONTENTS

INTRODUCTION

Our project is a virtual phonebook, based on the Yellow Pages and physical Telephone Directories that were commonly available. We intend to create a virtual version of the Telephone Directory, containing all possible means of contacting any given person.

The project allows the user to create contacts, modify or delete a particular contact, segregate them into seven different groups, search a specific person based on any detail, and also delete all contacts in one go. All the contacts are stored alphabetically, irrespective of the time they have been added. This has been done to ensure that they are easily readable by a person. The project has a user-friendly interface with easily recognisable icons for visual memory, if they are going to use it on a long-term basis. The application also has a dark mode to light mode switch, where the user can change the colour of the background, font colours, etc., depending on their need and liking. The project also records the timestamp at which the person has been added. This serves as a unique code for each person.

The project upon opening shows a menu containing all the different options. Upon pressing 'Create Contact', the person has to enter the name of the person, at the most three phone numbers, their email, date of birth, address, category, and finally, designation. All of these data will be stored in a nested list.

We first created the basic structure of the project using list and string functions, without any Graphical User Interface. Graphical User Interface was later on added to the project using Tkinter. Tkinter is one of the standard GUI libraries in Python. We have used Tkinter to add the background colour, message boxes, title, buttons, etc.

For the ease of programming and reading source code, we have divided the entire project into 4 different files and we have imported the functions of one file to another in order to access them. The different inbuilt modules used are sys, Datetime, Tkinter, etc.

IMPLEMENTATION

FILE 1:

```
#Dark Mode
from tkinter import *
from function import *
from tkinter import messagebox as mb


#"#7575a3" = light gray for labels
#"#3d3d5c" = dark gray for buttons
#"#9494b8" = bg
#"#1f1f2e"=form

#main window
root= Tk()
root.title("Phone Book")
root.geometry("1200x700")
root.minsize(1200,700)
root.iconbitmap("calltelephoneauricularincircularbutton_80086.ico")
root.configure(bg="#9494b8")

c_l=Label(text="copyright 2022",bg="#3d3d5c",
   fg="white",
   font=("Calibre", 10)
   ).pack(side=BOTTOM,fill=X,anchor=NW)

Title_label= Label(
    text="PHONEBOOK",relief=GROOVE,bg="#3d3d5c",fg="white",font=("Calibre", 30,"
bold"),borderwidth=10
).pack(side=TOP,fill=X)


welcome_label= Label(
    text="""Hello user, \n
Welcome to the smartphone phonebook""",
relief=SUNKEN,borderwidth=2,bg="#9494b8",fg="black",font=("Calibre", 10),pady=1,
).pack(anchor=N,side=TOP,fill=X)

create_img=PhotoImage(file="D:\\phone.py\icons8-add-96.png")
p = create_img.subsample(2,2)

create_contact=Button(root,
    text=" Create Contact",
    image=p,
    font=("Times New Roman",14,"bold"),
    command=createContact,
    compound=LEFT,
    bg="#3d3d5c",
    fg="white"
    ).pack(fill=X,pady=5)

all_img=PhotoImage(file="D:\\phone.py\icons8-contacts-80.png")
p1 = all_img.subsample(2,2)

all_contacts=Button(
    root,
    text=" All Contacts",
    font=("Times New Roman",14,"bold"),
    command=allContacts,
    bg="#3d3d5c",
    fg="white",
    image=p1,
    compound=LEFT,
    ).pack(fill=X,pady=5)

search_img=PhotoImage(file="D:\\phone.py\icons8-search-128.png")
p2 = search_img.subsample(3,3)

search_contact=Button(
    root,
    text=" Search Contact",
    font=("Times New Roman",14,"bold"),
    command=searchContact,
    bg="#3d3d5c",
    fg="white",
    image=p2,
    compound=LEFT,
    ).pack(fill=X,pady=5)

group_img=PhotoImage(file="D:\\phone.py\icons8-group-96.png")
p3 = group_img.subsample(2,2)

see_groups=Button(
    root,
    text="  Groups",
    font=("Times New Roman",14,"bold"),
    command=manageGroups,
    bg="#3d3d5c",
    fg="white",
    image=p3,
    compound=LEFT,
    ).pack(fill=X,pady=5)

dummy_img=PhotoImage(file="D:\\phone.py\icons8-data-sheet-80.png")
p4 = dummy_img.subsample(2,2)

data_dummy=Button(
    root,
    text=" Add Sample Data",
    font=("Times New Roman",14,"bold"),
    command=dummyData,
    bg="#3d3d5c",
    fg="white",
    image=p4,
    compound=LEFT,
    ).pack(fill=X,pady=5)



delete_img=PhotoImage(file="D:\\phone.py\icons8-delete-300.png")
p5 = delete_img.subsample(6,6)

delete_contact=Button(
    root,
    text=" Delete All Contacts",
    command=deleteallContact,
    font=("Times New Roman",14,"bold"),
    bg="#3d3d5c",
    fg="white",
    image=p5,
    compound=LEFT,
    ).pack(fill=X,pady=5)

exit_img=PhotoImage(file="D:\\phone.py\icons8-exit-256.png")
p6 = exit_img.subsample(6,6)

exit_prg=Button(
    root,
    text="  Exit",
    font=("Times New Roman",14,"bold"),
    command=exit_prg,
    bg="#3d3d5c",
    fg="white",
    image=p6,
    compound=LEFT,
    ).pack(fill=X,pady=5)

mode_prg=Button(
    root,
    text=" Convert to Light Mode",
    font=("Times New Roman",14,"bold"),
    command=lambda:[root.destroy(),light_mode()],
    bg="#3d3d5c",
    fg="white",
    ).pack(fill=X,pady=5)

root.mainloop()
```

FILE 2:

```
#Light Mode

from tkinter import *
from function2 import *
from tkinter import messagebox as mb

#"#CBF7F3" = heading labels
#"#9ccefa" = buttons
#"#CED9F7" = bg
#"#c3e8e2"=form
#"#b8cde0"= Ask Skanda
root= Tk() #main window
root.title("Phone Book")
root.geometry("1200x700")
root.minsize(1200,700)
root.iconbitmap("calltelephoneauricularincircularbutton_80086.ico")
root.configure(bg="#CED9F7")

c_l=Label(text="Copyright 2022",bg="#b8cde0",
    fg="black",
    font=("Times New Roman", 10)
    ).pack(side=BOTTOM,fill=X,anchor=NW)

Title_label= Label(
    text="Phonebook", relief=GROOVE,bg="#CBF7F3",fg="black",font=("Times New Roman", 30,"
bold"),borderwidth=10
```

```
).pack(side=TOP,fill=X)


welcome_label= Label(
    text="""Hello user, \n
Welcome to the Smartphone Phonebook""",
relief=SUNKEN,borderwidth=2,bg="#CED9F7",fg="black",font=("Times New Roman", 10),pady=1,
).pack(anchor=N,side=TOP,fill=X)

create_img=PhotoImage(file="D:\phone.py\icons8-add-96.png")
p = create_img.subsample(2,2)

create_contact=Button(root,
    text=" Create Contact",
    image=p,
    font=("Times New Roman",14,"bold"),
    command=createContact,
    compound=LEFT,
    bg="#9ccefa",
    fg="black"
    ).pack(fill=X,pady=5)

all_img=PhotoImage(file="D:\phone.py\icons8-contacts-80.png")
p1 = all_img.subsample(2,2)

all_contacts=Button(
    root,
    text="        All Contacts",
    font=("Times New Roman",14,"bold"),
    command=allContacts,
    bg="#9ccefa",
    fg="black",
    image=p1,
    compound=LEFT,
    ).pack(fill=X,pady=5)

search_img=PhotoImage(file="D:\phone.py\icons8-search-128.png")
p2 = search_img.subsample(3,3)

search_contact=Button(
    root,
    text=" Search Contact",
    font=("Times New Roman",14,"bold"),
    command=searchContact,
    bg="#9ccefa",
    fg="black",
    image=p2,
    compound=LEFT,
    ).pack(fill=X,pady=5)

group_img=PhotoImage(file="D:\phone.py\icons8-group-96.png")
p3 = group_img.subsample(2,2)

see_groups=Button(
    root,
    text="                 Groups",
    font=("Times New Roman",14,"bold"),
    command=manageGroups,
    bg="#9ccefa",
    fg="black",
    image=p3,
    compound=LEFT,
    ).pack(fill=X,pady=5)

dummy_img=PhotoImage(file="D:\phone.py\icons8-data-sheet-80.png")
p4 = dummy_img.subsample(2,2)

data_dummy=Button(
    root,
    text="    Add Sample Data",
    font=("Times New Roman",14,"bold"),
    command=dummyData,
    bg="#9ccefa",
    fg="black",
    image=p4,
    compound=LEFT,
    ).pack(fill=X,pady=5)


delete_img=PhotoImage(file="D:\phone.py\icons8-delete-300.png")
p5 = delete_img.subsample(6,6)

delete_contact=Button(
    root,
    text="      Delete All Contacts",
    command=deleteallContact,
    font=("Times New Roman",14,"bold"),
    bg="#9ccefa",
    fg="black",
    image=p5,
    compound=LEFT,
    ).pack(fill=X,pady=5)

exit_img=PhotoImage(file="D:\phone.py\icons8-exit-256.png")
p6 = exit_img.subsample(6,6)

exit_prg=Button(
    root,
    text="             Exit",
    font=("Times New Roman",14,"bold"),
    command=exit_prg,
    bg="#9ccefa",
    fg="black",
    image=p6,
    compound=LEFT,
    ).pack(fill=X,pady=5)

mode_prg=Button(
    root,
    text=" Convert to Dark Mode",
    font=("Times New Roman",14,"bold"),
    command=lambda:[root.destroy(),dark_mode()],
    bg="#9ccefa",
    fg="black",
    ).pack(fill=X,pady=5)

root.mainloop()
```

FILE 3:

```
from tkinter import *
from tkinter import messagebox as mb
from tkinter import ttk
from types import NoneType
import datetime
import sys

phonebook=[]


#for creating Tkobject
def object(t,g,min,bg):
    r=Tk()
    a,b=min
    r.title(t)
    r.geometry(g)
    r.iconbitmap("calltelephoneauricularincircularbutton_80086.ico")
    r.minsize(a,b)
    r.configure(bg=bg)
    return r


#createContact is used to create new contact.
def createContact():
    r= object("Create Contact", "1100x600",(1100,600),"#7575a3")
    Title= Label(r,
    text="Create Contact",
relief=GROOVE,bg="#3d3d5c",fg="white",font=("Calibri",30,"bold"),borderwidth=10
    ).pack(side=TOP,fill=X)

    name_var=StringVar(master=r)
    phone_var=StringVar(master=r)
    phone2_var=StringVar(master=r)
    phone3_var=StringVar(r)
    email_var=StringVar(r)
    dob_var=StringVar(r)
    group_var=StringVar(r)
    profession_var=StringVar(r)
    address_var=StringVar(r)

    def submit():
        c=0
        unique_id = str(datetime.datetime.now())
        name=name_var.get()
        phone=phone_var.get()



        if phone2_var!=NoneType:
            phone2=phone2_var.get()

            phone2_var.set("")
        else:
            phone2=""
        if phone3_var!=NoneType:
            phone3=phone3_var.get()

            phone3_var.set("")
```

```
else:
    phone3=""
email=email_var.get()
dob=dob_var.get()
group=group_var.get()
address =address_var.get()
profession= profession_var.get()
#validating
if phone!="":
    if not(phone.isdigit()):
        mb.showerror("Number can contain only digits","Number can contain only digits")
        r.destroy()
        c=1
if email!="":
    if not('@' in email):
        mb.showerror("Invalid Email","Invalid Email")
        r.destroy()
        c=1

if phone2 !="":
    if not(phone2.isdigit()):
        mb.showerror("Number can contain only digits","Number can contain only digits")
        r.destroy()
        c=1

if phone3 !="":
    if not(phone3.isdigit()):
        mb.showerror("Number can contain only digits","Number can contain only digits")
        r.destroy()
```

```
            c=1

    if dob!="":
        if len(dob)!=10 and not("/" in dob):
            mb.showerror("Invalid Date of Birth Format","Invalid Date of Birth Format")
            r.destroy()
            c=1
    if profession!="":
        if not profession.isalpha():
            mb.showerror("Profession Can't contain digits or special characters","Profession Can't
contain digits or special characters")
            r.d estroy()
            c=1
    if c==0:

phonebook.append([name,phone,phone2,phone3,email,dob,group,address,profession,unique_id])
        mb.showinfo("Contact Created", "The Contact is Successfully Created")
        r.destroy()

    name_var.set("")
    phone_var.set("")
    email_var.set("")
    dob_var.set("")
    group_var.set("")
    address_var.set("")
    profession_var.set("")

    print(phonebook)
```

```
def add2ndnum():
    nonlocal phone2_var

    phone2_label=Label(r,
    text="Number 2:",
    font =("Times New Roman",15),
    bg="#1f1f2e",
    fg="white",
    ).pack(side=TOP,anchor=N,fill=X,padx=100)
    phone2_entry=Entry(r,width=60,textvariable=phone2_var,).pack(side=TOP,anchor=N,padx=100)

    add=Button(r,
    text="ADD THIRD NUMBER",
    command=add3rdnum,
    font=("Times New Roman",15,"bold"),
    bg="#3d3d5c",
    fg="white",).pack(side=TOP,anchor=N,fill=X,padx=200)
```

```
def add3rdnum():
    nonlocal phone3_var


    phone3_label=Label(r,
    text="Number 3:",
    font =("Times New Roman", 15),
    bg="#1f1f2e",
    fg="white",
    ).pack(side=TOP,anchor=N,fill=X,padx=100)
    phone3_entry=Entry(r,width=60,textvariable=phone3_var,).pack(side=TOP,anchor=N,padx=100)

sub=Button(r,
    text="Submit",
    command=submit,
    bg="#3d3d5c",
    font=("Times New Roman",15,"bold"),
    pady=8,
    fg="white",
    relief=GROOVE,
    )

sub.pack(side=TOP,anchor=N, fill=X,padx=200,pady=9)
```

```
    phone2_var.set("")
    phone3_var.set("")

    name_label=Label(r,
    text="Name:",
    font =("Times New Roman",15),
    bg="#1f1f2e",
    fg="white",
    ).pack(side=TOP,anchor=N,fill=X,padx=100)
    name_entry=Entry(r,width=100,textvariable=name_var,font = ('Times New
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100)

    phone_label=Label(r,
    text="Number 1:",
    font =("Times New Roman",15),
    bg="#1f1f2e",
    fg="white",
    ).pack(side=TOP,anchor=N,fill=X,padx=100)
    phone_entry=Entry(r,width=60,textvariable=phone_var,font = ('Times New
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100)
    add=Button(r,
        text="ADD SECOND NUMBER",
        command=add2ndnum,
        font=("Times New Roman",15,"bold"),
        bg="#3d3d5c",
        fg="white",).pack(side=TOP,anchor=N,fill=X,padx=200)

    email_label=Label(r,text="Email:",
        font =("Times New Roman",15),
        bg="#1f1f2e",
        fg="white",
        ).pack(side=TOP,anchor=N,fill=X,padx=100)
    email_entry=Entry(r,width=100,textvariable=email_var,font = ('Times New
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100)

    dob_label=Label(r,text="Date of birth(DD/MM/YYYY):",
        font =("Times New Roman",15),
        bg="#1f1f2e",
        fg="white",
        ).pack(side=TOP,anchor=N,fill=X,padx=100)
    dob_entry=Entry(r,width=50,textvariable=dob_var,font = ('Times New
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100)



category_label=Label(r,text="Category(Family/Friends/Work/Favorite/Emergency/Important/Others):"
,
        font =("Times New Roman",15),
        bg="#1f1f2e",
        fg="white",
        ).pack(side=TOP,anchor=N,fill=X,padx=100)
    category_entry=Entry(r,width=60,textvariable=group_var,font = ('Times New
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100)


    address_label=Label(r,text="Address:",
        font =("Times New Roman",15),
        bg="#1f1f2e",
        fg="white",
        ).pack(side=TOP,anchor=N,fill=X,padx=100)
    address_entry=Entry(r,width=60,textvariable=address_var,font = ('Times New
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100)

    profession_label=Label(r,text="Designation :",
        font =("Times New Roman",15),
        bg="#1f1f2e",
        fg="white",
        ).pack(side=TOP,anchor=N,fill=X,padx=100)
    profession_entry=Entry(r,width=60,textvariable=profession_var,font = ('Times New
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100)

    r.mainloop()

#editing the contact
def editContact(i,t,t1):
    r= object("Edit Contact", "1100x600",(1100,600),"#7575a3")
    Title= Label(r,
    text="Edit Contact",
    relief=GROOVE,bg="#3d3d5c",fg="white",font=("Calibri",30,"bold"),borderwidth=
    10
    ).pack(side=TOP,fill=X)
    i[1]=str(i[1])
    i[2]=str(i[2])
    i[3]=str(i[3])
    name_var=StringVar(master=r)
    phone_var=StringVar(master=r)
    phone2_var=StringVar(master=r)
```

```
phone3_var=StringVar(r)
email_var=StringVar(r)
dob_var=StringVar(r)
group_var=StringVar(r)
profession_var=StringVar(r)
address_var=StringVar(r)

def submit():
    c=0
    unique_id = i[9]
    name=name_var.get()
    phone=phone_var.get()

    if phone2_var!=NoneType:
        phone2=phone2_var.get()

        phone2_var.set("")
    else:
        phone2=""
    if phone3_var!=NoneType:
        phone3=phone3_var.get()

        phone3_var.set("")
    else:
        phone3=""
    email=email_var.get()
    dob=dob_var.get()
    group=group_var.get()
    address =address_var.get()
```

```
profession= profession_var.get()
d=phonebook.index(i)

#validating
if phone!="":
    if not(phone.isdigit()):
        mb.showerror("Number can contain only digits","Number can contain only digits")
        r.destroy()
        c=1
if email!="":
    if not('@' in email):
        mb.showerror("Invalid Email","Invalid Email")
        r.destroy()
        c=1

if phone2 !="":
    if not(phone2.isdigit()):
        mb.showerror("Number can contain only digits","Number can contain only digits")
        r.destroy()
        c=1

if phone3 !="":
    if not(phone3.isdigit()):
        mb.showerror("Number can contain only digits","Number can contain only digits")
        r.destroy()
        c=1

if dob!="":
    if len(dob)!=10 and not("/" in dob):
            mb.showerror("Invalid Date of Birth Format","Invalid Date of Birth Format")
            r.destroy()
            c=1
    if profession!="":
        if not profession.isalpha():
            mb.showerror("Profession Can't contain digits or special characters","Profession Can't
            contain digits or special characters")
            r.destroy()
            c=1

    if c==0:
        phonebook[d]=[name,phone,phone2,phone3,email,dob,group,address,profession,unique_id]



        print(phonebook)

        mb.showinfo("Contact Updated", "The Contact is Successfully Updated")
        r.destroy()
    t.destroy()
    t1.destroy()




sub=Button(r,
    text="Update Contact",
    command=submit,
    bg="#3d3d5c",
    font=("Times New Roman",15,"bold"),
    pady=8,
  fg="white",
  relief=GROOVE,).pack(side=TOP,anchor=N,padx=200,pady=9)

name_var.set(i[0])
phone_var.set(i[1])
phone2_var.set(i[2])
phone3_var.set(i[3])
email_var.set(i[4])
dob_var.set(i[5])
group_var.set(i[6])
address_var.set(i[7])
profession_var.set(i[8])

name_label=Label(r,
text="Name:",
font =("Times New Roman",15),
bg="#1f1f2e",
fg="white",
).pack(side=TOP,anchor=N,fill=X,padx=100)
name_entry=Entry(r,width=100,textvariable=name_var,font = ('Times New
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=1
00) phone_label=Label(r,
text="Number 1:",
font =("Times New Roman",15),
bg="#1f1f2e",
fg="white",
).pack(side=TOP,anchor=N,fill=X,padx=100)
phone_entry=Entry(r,width=60,textvariable=phone_var,font = ('Times New
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100)
```

```
phone2_label=Label(r,
    text="Number 2:",
    font =("Times New Roman",15),
    bg="#1f1f2e",
    fg="white",
    ).pack(side=TOP,anchor=N,fill=X,padx=100)
phone2_entry=Entry(r,width=60,textvariable=phone2_var,).pack(side=TOP,anchor=N,padx=100)
phone3_label=Label(r,
    text="Number 3:",
    font =("Times New Roman",15),
    bg="#1f1f2e",
    fg="white",
    ).pack(side=TOP,anchor=N,fill=X,padx=100)
phone3_entry=Entry(r,width=60,textvariable=phone3_var,).pack(side=TOP,anchor=N,padx=100)
email_label=Label(r,text="Email:",
    font =("Times New Roman",15),
    bg="#1f1f2e",
    fg="white",
    ).pack(side=TOP,anchor=N,fill=X,padx=100)
email_entry=Entry(r,width=100,textvariable=email_var,font = ('Times New
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100
) dob_label=Label(r,text="Date of
birth(DD/MM/YYYY):",
    font =("Times New Roman",15),
    bg="#1f1f2e",
    fg="white",
    ).pack(side=TOP,anchor=N,fill=X,padx=100)
dob_entry=Entry(r,width=60,textvariable=dob_var,font = ('Times New
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100)
```

```
category_label=Label(r,text="Category(Family/Friends/Work/Favorite/Emergency/Important/Others):"
,
        font =("Times New Roman",15),
        bg="#1f1f2e",
        fg="white",
        ).pack(side=TOP,anchor=N,fill=X,padx=100)
    category_entry=Entry(r,width=60,textvariable=group_var,font = ('Times New
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100)

    address_label=Label(r,text="Address:",
        font =("Times New Roman",15),
        bg="#1f1f2e",
        fg="white",
        ).pack(side=TOP,anchor=N,fill=X,padx=100)
    address_entry=Entry(r,width=100,textvariable=address_var,font = ('Times New
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100)
    profession_label=Label(r,text="Designation :",
        font =("Times New Roman",15),
        bg="#1f1f2e",
        fg="white",
        ).pack(side=TOP,anchor=N,fill=X,padx=100)
    profession_entry=Entry(r,width=60,textvariable=profession_var,font = ('Times New
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100)

    r.mainloop()


def delete(r1,r,i):
```

```
#i=phonebook.index(i)
r2=object( "{} Contact Deleted".format(i[0]), "300x100",(300,100),"#7575a3")
r2.maxsize(300,100)
i[1]=str(i[1])
i[2]=str(i[2])
i[3]=str(i[3])

phonebook.remove(i)

Label(r2,text="Contact Deleted",padx=10,pady=10,bg="#1f1f2e",
font=("Times New Roman",15),
fg="white",relief=GROOVE).pack(side=TOP,anchor=CENTER,fill=X)

r1.destroy()
r.destroy()

Button(
    r2,
    text="OK",
    font=("Times New Roman",15,"bold"),
    bg="#1f1f2e",
    fg="white",
    command=r2.destroy
    ).pack(side=BOTTOM,anchor=S,fill=X,padx=10,pady=10)



r2.mainloop()
```

```
def open(i,r):
   r1=object( "{} Contact Info".format(i[0]), "500x500",(500,500),"#7575a3")
   Title= Label(r1,
   text="Contact Details",
relief=GROOVE,bg="#3d3d5c",fg="white",font=("Calibri",30,"bold"),borderwidth=10
      ).pack(side=TOP,fill=X)

   Label(master=r1,
      text="Name : "+ i[0],
      font =("Times New Roman",15),
      bg="#1f1f2e",
      fg="white",
      relief=SUNKEN,
      borderwidth=2,
      padx=10,
         pady=3).pack(fill=X,side=TOP)

   Label(r1,
         text="Number :  " +str(i[1]),
         font =("Times New Roman",15),
         bg="#1f1f2e",
         fg="white",
         relief=SUNKEN,
         borderwidth=2,
         padx=10,
         pady=3,
         ).pack(fill=X,side=TOP)

   if str(i[2])!="":
    Label(r1,
            text="Number 2 : " +str(i[2]),
            font =("Times New Roman",15),
            bg="#1f1f2e",
            fg="white",
            relief=SUNKEN,
            borderwidth=2,
            padx=10,
            pady=3,
            ).pack(fill=X,side=TOP)

if str(i[3])!="":
    Label(r1,
            text="Number 3 : " +str(i[3]),
            font =("Times New Roman",15),
            bg="#1f1f2e",
            fg="white",
            relief=SUNKEN,
            borderwidth=2,
            padx=10,
            pady=3,
            ).pack(fill=X,side=TOP)

Label(r1,
        text="Email : " +str(i[4]),
        font =("Times New Roman",15),
        bg="#1f1f2e",
        fg="white",
        relief=SUNKEN,
        borderwidth=2,
        padx=10,
        pady=3,
        ).pack(fill=X,side=TOP)

Label(r1,
        text="Date of Birth : " +str(i[5]),
        font =("Times New Roman",15),
        bg="#1f1f2e",
        fg="white",
        relief=SUNKEN,
        borderwidth=2,
        padx=10,
        pady=3,
        ).pack(fill=X,side=TOP)


Label(r1,
        text="Group : " +i[6],
        font =("Times New Roman",15),
        bg="#1f1f2e",
        fg="white",
        relief=SUNKEN,
        borderwidth=2,
        padx=10,
        pady=3,
        ).pack(fill=X,side=TOP)

Label(r1,
        text="Address :  " +str(i[7]),
        font =("Times New Roman",15),
        bg="#1f1f2e",
        fg="white",
        relief=SUNKEN,
        borderwidth=2,
        padx=10,
        pady=3,
        ).pack(fill=X,side=TOP)

Label(r1,
        text="Designation  :  "  +str(i[8]),
        font =("Times New Roman",15),
        bg="#1f1f2e",
        fg="white",
        relief=SUNKEN,
        borderwidth=2,
        padx=10,
        pady=3,
        ).pack(fill=X,side=TOP)

Label(r1,
        text="Timestamp : " +str(i[9]),
        font =("Times New Roman",15),
        bg="#1f1f2e",
        fg="white",
        relief=SUNKEN,
        borderwidth=2,
        padx=10,
        pady=3,
        ).pack(fill=X,side=TOP)



""""
edit_img=PhotoImage(r1,file="D:\phone.py\icons8-edit-40.png")
p6 = edit_img.subsample(1,1)

delete1_img=PhotoImage(r1,file="D:\phone.py\icons8-delete-300.png")
p7 = delete1_img.subsample(6,6)
""""
g_img=PhotoImage(master=r1,file="D:\phone.py\icons8-delete-300.png")
i1 = g_img.subsample(6,6)

g1_img=PhotoImage(master=r1,file="D:\phone.py\icons8-edit-80.png")
i2 = g1_img.subsample(2,2)

Button(r1,
        text=" Edit Contact",
        command=lambda:[editContact(i,r,r1)],
        font=("Times New Roman",15),
        fg="white",
        bg="#3d3d5c",
        pady=5,
        image=i2,
        compound=LEFT,
        ).pack(side=BOTTOM,fill=X)
```

```
    Button(r1,text= " Delete Contact",
            #way to send parameters
            #command=lambda:delete(r1),
            command=lambda:[delete(r1,r,i)],
            font=("Times New Roman",15),
            fg="white",
            bg="#3d3d5c",
            pady=5,
            #image=p7,
            #compound=LEFT,
            image=i1,
            compound=LEFT,
            ).pack(side=BOTTOM,fill=X)


    r1.mainloop()

def openList(l):
    r=object("Contacts","1100x600",(1100,600),"#7575a3")

    Title= Label(r,
    text="Contacts",
relief=GROOVE,bg="#3d3d5c",fg="white",font=("Calibri",30,"bold"),borderwidth=10
    ).pack(side=TOP,fill=X)

    create_img=PhotoImage(master=r,file="D:\phone.py\icons8-add-96.png")
    p = create_img.subsample(2,2)

    create_contact=Button(r,
    text=" Create Contact",
    image=p,
    font=("Times New Roman",14,"bold"),
    command=createContact,
    compound=LEFT,
    bg="#3d3d5c",
    fg="white"
    ).pack(padx=200,fill=X,pady=2)

    search_var=StringVar(r)

    def submit():
        global phonebook
        q=search_var.get()
        print(q,type(q))
        l=[]
        c=0
        for i in phonebook:

        print(i)

            if i[0].startswith(q) or i[0].endswith(q) or i[1]==q or i[2]==q or i[3]==q or i[4]==q or i[6]==q or i[7].startswith(q) or i[7].endswith(q) or i[8]==q:
                    l.append(i)
                    c=1

            print("c=",c)
        if c==0:
            mb.showinfo("No Contact Found","No results found")
        elif c==1:
            r.destroy()
            openList(l)



    search_label=Label(r,text="Search :",
        font =("Times New Roman",15,"bold"),
        bg="#1f1f2e",
        fg="white",
        ).pack(side=TOP,anchor=N,fill=X,pady=4)
    search_entry=Entry(r,textvariable=search_var,font = ('Times New
Roman',10,'normal'),width=100).pack(side=TOP,anchor=N)

    g1_img=PhotoImage(master=r,file="D:\phone.py\icons8-search-96.png")
    i2 = g1_img.subsample(3,3)

    Button(r,
        text=" Search",
        command=submit,
        bg="#3d3d5c",
        fg="white",
        font=("Times New Roman",15,"bold"),
        padx=10,
        pady=5,
        image=i2,
        compound=LEFT
        ).pack(side=TOP,anchor=N,padx=200,fill=X,pady=2)
```

```
scroll_bar = Scrollbar(r)

scroll_bar.pack( side = RIGHT,
          fill = Y )

columns = ('name', 'phone_number_1',"phone_number_2","phone_number_3",
'email',"date_of_birth","group","address","profession","unique_id")
tree = ttk.Treeview(r, columns=columns, show='headings',yscrollcommand = scroll_bar.set)
tree.heading("name",text="Name")
tree.heading("phone_number_1",text="Phone Number 1")
tree.heading("phone_number_2",text="Phone Number 2")
tree.heading("phone_number_3",text="Phone Number 3")
tree.heading("email",text="Email")
tree.heading("date_of_birth",text="Date of Birth")
tree.heading("group",text="Group")
tree.heading("address",text="Address")
tree.heading("profession",text="Profession")
tree.heading("unique_id",text="Time Stamp")

tree.column('name', width=60, anchor=W)
tree.column('phone_number_1', width=15,)
tree.column('phone_number_2', width=15,)
tree.column('phone_number_3', width=15, )
tree.column('email', width=60, )
```

```
    tree.column('date_of_birth',width=8,)
    tree.column('group', width=20, )
    tree.column('address', width=50, )
    tree.column('profession', width=10,)
    tree.column('unique_id', width=10,)

    l.sort(key=lambda x:x[0])

    for i in l:
        tree.insert('', END, values=i)

    def item_selected(event):
        for selected_item in tree.selection():
            item = tree.item(selected_item)
            record = item['values']
            print(record)
            open(record,r)

    tree.bind('<Double-1>', item_selected)

    tree.pack(fill=BOTH,expand=True)

    #scrollbar = ttk.Scrollbar(r, orient=VERTICAL, command=tree.yview)
    #tree.configure(yscroll=scrollbar.set)
    #scrollbar.pack(side=LEFT,fill=Y)

    r.mainloop()

def dummyData():
    #phonebook.append([name,phone,phone2,phone3,email,dob,group,address,profession,unique_id])
    #Dummy data is to add entries quickly so that testing can be done quickly
    phonebook.append(["Police","100","","","gov@gmail.com","","emergency","Ranchi Jharkhand","police",str(datetime.datetime.now())])
    phonebook.append(["Hospital","112","","","hospital@gmail.com","","emergency","Ranchi Jharkhand","hospital",str(datetime.datetime.now())])
    phonebook.append(["Shreya Mishra","9193939999","9343387733","9876543210","shreya@gmail.com","22/02/2002","friend","Ranchi Jharkhand","student",str(datetime.datetime.now())])
    phonebook.append(["Apoorva Raj","9193933499","9343387732","9126543210","apoorva@gmail.com","25/12/2002","important","Mysore Karnataka","student",str(datetime.datetime.now())])
    phonebook.append(["Adrija Pes","9419393559","9343456733","9876733210","adrija@gmail.com","05/05/2004","favourite","Bangalore Karnataka","student",str(datetime.datetime.now())])
    phonebook.append(["Abhijeet Mishra","9193456999","9999387733","9873293210","abhijeet@gmail.com","09/11/1978","work","Hyderabad Telangana","teacher",str(datetime.datetime.now())])
    phonebook.append(["Isham Sinha","9567456999","9934587733","9878943210","isham@gmail.com","09/01/2007","friend","Bangalore Karnataka","student",str(datetime.datetime.now())])
    phonebook.append(["B. S. Grewal","9193456999","9999387733","9873293210","grewal@gmail.com","07/08/2002","classmate","Hyderabad Telangana","teacher",str(datetime.datetime.now())])
    phonebook.append(["Skanda Sreesha Prashad","9193402399","9991234733","9870912210","skanda@gmail.com","04/02/2005","friend","Bangalore Karnataka","student",str(datetime.datetime.now())])
    phonebook.append(["Chandan Kumar Jha","9155402399","9991111133","9222912210","chandan@gmail.com","04/12/1985","family","Bangalore Karnataka","student",str(datetime.datetime.now())])
    phonebook.append(["Rashmi Jha","9155433399","9333111133","9233312210","rashmi@gmail.com","02/07/1985","family","Bangalore Karnataka","student",str(datetime.datetime.now())])
    mb.showinfo("Data added", "Dummy Data Entries is added to the phone directory")

def searchContact():
    global phonebook

    r= object("Search Contact", "600x600",(600,600),"#7575a3")
    Title= Label(r,
    text="Search Contact",
relief=GROOVE,bg="#3d3d5c",fg="white",font=("Calibri",30,"bold"),borderwidth=10
    ).pack(side=TOP,fill=X)

    search_var=StringVar(r)

    def submit():
        global phonebook
        q=search_var.get()
        print(q,type(q))
        l=[]
        c=0
        for i in phonebook:
            #print(i)
        if i[0].startswith(q) or i[0].endswith(q) or i[1]==q or i[2]==q or i[3]==q or i[4]==q or i[6]==q or
i[7].startswith(q) or i[7].endswith(q) or i[8]==q:
            l.append(i)
            c=1

        print("c=",c)
      if c==0:
        mb.showinfo("No Contact Found","No results found")
      elif c==1:
        openList(l)


    search_label=Label(r,text="Search :",
      font =("Times New Roman",15,"bold"),
      bg="#1f1f2e",
      fg="white",
      ).pack(side=TOP,anchor=N,fill=X,pady=4)
    search_entry=Entry(r,textvariable=search_var,font = ('Times New
Roman',10,'normal'),width=100).pack(side=TOP,anchor=N)

    g1_img=PhotoImage(master=r,file="D:\phone.py\icons8-search-96.png")
    i2 = g1_img.subsample(3,3)

    Button(r,
      text="Search",
      command=submit,
      bg="#3d3d5c",
      fg="white",
      font=("Times New Roman",15,"bold"),
        padx=10,
        pady=5,
        image=i2,
        compound=LEFT
        ).pack(side=TOP,anchor=N,padx=200,fill=X,pady=10)


    r.mainloop()



def allContacts():
    openList(phonebook)


def manageGroups():
    r=object("Groups","600x600",(600,600),"#7575a3")
    Title= Label(r,
    text="Groups",
    relief=GROOVE,bg="#3d3d5c",fg="white",font=("Calibri",30,"bold"),
    borderwidth=10
    ).pack(side=TOP,fill=X)
    def exist():
        t=["work","friends","friend","family","emergency","favourite","important"]
        l=[]

        for i in phonebook:
            if i[6].lower() not in t:
                l.append(i[6])

        return l
```

```
def work():
    l=[]
    for i in phonebook:
        if i[6]=="Work" or i[6]=="WORK" or i[6]=="work":
            l.append(i)
    if l!=[]:
        openList(l)
    else:
        mb.showinfo("Empty Group","No one present in the group")

def friend():
    l=[]
    for i in phonebook:
        if i[6]=="Friends" or i[6]=="FRIENDS" or i[6]=="friends" or i[6]=="friend" or i[6]=="Friend"
or i[6]=="FRIEND":
            l.append(i)
    if l!=[]:
        openList(l)
    else:
        mb.showinfo("Empty Group","No one present in the group")

def family():
    l=[]
    for i in phonebook:
        if i[6]=="Family" or i[6]=="FAMILY" or i[6]=="family":
            l.append(i)
    if l!=[]:
        openList(l)
    else:
        mb.showinfo("Empty Group","No one present in the group")

def favourite():
    l=[]
    for i in phonebook:
        if i[6]=="favourite" or i[6]=="Favourite" or i[6]=="favourite".upper():
            l.append(i)
    if l!=[]:
        openList(l)
    else:
        mb.showinfo("Empty Group","No one present in the group")

def important():
    l=[]
    for i in phonebook:
        if i[6]=="important" or i[6]=="Important" or i[6]=="important".upper():
            l.append(i)
    if l!=[]:
        openList(l)
    else:
        mb.showinfo("Empty Group","No one present in the group")

def emergency():
    l=[]
    for i in phonebook:
        if i[6]=="emergency" or i[6]=="Emergency" or i[6]=="EMERGENCY":
            l.append(i)
    if l!=[]:
        openList(l)
    else:
        mb.showinfo("Empty Group","No one present in the group")

def others():
    l=[]
    a=exist()
    for i in phonebook:
        if i[6] in a:
            l.append(i)

    if l!=[]:
        openList(l)
    else:
        mb.showinfo("Empty Group","Noone present in the group")

g_img=PhotoImage(master=r,file="D:\phone.py\icons8-emergency-64.png")
i = g_img.subsample(2,2)

g1_img=PhotoImage(master=r,file="D:\phone.py\icons8-work-64.png")
i1 = g1_img.subsample(2,2)

g2_img=PhotoImage(master=r,file="D:\phone.py\icons8-friends-96.png")
i2 = g2_img.subsample(2,2)

g3_img=PhotoImage(master=r,file="D:\phone.py\icons8-family-120.png")
i3 = g3_img.subsample(3,3)

g4_img=PhotoImage(master=r,file="D:\phone.py\icons8-group-96 (1).png")
```

```
i4 = g4_img.subsample(2,2)

g5_img=PhotoImage(master=r,file="D:\phone.py\icons8-favourite-64.png")
i5 = g5_img.subsample(2,2)

g6_img=PhotoImage(master=r,file="D:\phone.py\icons8-important-64.png")
i6 = g6_img.subsample(2,2)



Button(
    r,
    text=" Emergency Contacts",
    command=emergency,
    bg="#3d3d5c",
    fg="white",
    font =("Times New Roman",15),
    compound=LEFT,
    image=i

    ).pack(fill=X,pady=5)

Button(
    r,
    text=" Important Contacts",
    command=important,
    bg="#3d3d5c",
    fg="white",
    font =("Times New Roman",15),
    compound=LEFT,
    image=i6

    ).pack(fill=X,pady=5)

Button(
    r,
    text=" Favourite Contacts",
    command=favourite,
    bg="#3d3d5c",
    fg="white",
    font =("Times New Roman",15),
    compound=LEFT,
    image=i5

    ).pack(fill=X,pady=5)

Button(
    r,
    text=" Work",
    command=work,
    bg="#3d3d5c",
    fg="white",
    font =("Times New Roman",15),
    compound=LEFT,
    image=i1

    ).pack(fill=X,pady=5)
```

```
Button(
    r,
    text=" Friends",
    command=friend,
    bg="#3d3d5c",
    fg="white",
    font =("Times New Roman",15),
    compound=LEFT,
    image=i2
    ).pack(fill=X,pady=5)
Button(
    r,
    text=" Family",
    command=family,
    bg="#3d3d5c",
    fg="white",
    font =("Times New Roman",15),
    compound=LEFT,
    image=i3
    ).pack(fill=X,pady=5)
Button(
    r,
    text=" Others",
    command=others,
    bg="#3d3d5c",
    fg="white",
    font =("Times New Roman",15),
    compound=LEFT,
    image=i4
```

```
        ).pack(fill=X,pady=5)

    r.mainloop()

def deleteallContact():
    global phonebook
    phonebook=[]
    print(phonebook)
    mb.showinfo("All contacts Deleted","All Contacts Deleted")

def exit_prg():
    sys.exit(0)


def setPhonebook(pb):
    global phonebook
    phonebook =pb

def light_mode():
    import function2
    function2.setPhonebook(phonebook)
    import main2
```

FILE 4:

```
from tkinter import *
from tkinter import messagebox as mb
from tkinter import ttk
from types import NoneType
import datetime
import sys

#from tkinter.ttk import *

phonebook=[]



def object(t,g,min,bg):
   r=Tk()
   a,b=min
   r.title(t)
   r.geometry(g)
   r.iconbitmap("calltelephoneauricularincircularbutton_80086.ico")
   r.minsize(a,b)
   r.c onfigure(bg=bg)
   return r

def createContact():
   r= object("Create Contact", "1100x600",(1100,600),"#CED9F7")
   Title= Label(r,
   text="Create Contact", relief=GROOVE,bg="#CBF7F3",fg="black",font=("Times New
Roman",30,"bold"),borderwidth=10
   ).pack(side=TOP,fill=X)

   name_var=StringVar(master=r)
   phone_var=StringVar(master=r)
   phone2_var=StringVar(master=r)
   phone3_var=StringVar(r)
   email_var=StringVar(r)
   dob_var=StringVar(r)
   group_var=StringVar(r)
   profession_var=StringVar(r)
   address_var=StringVar(r)

   def submit():
       c=0
       unique_id = str(datetime.datetime.now())
       name=name_var.get()
       phone=phone_var.get()

       if phone2_var!=NoneType:
           phone2=phone2_var.get()
           phone2_var.set("")
       else:
           phone2=""

       if phone3_var!=NoneType:
           phone3=phone3_var.get()

           phone3_var.set("")
       else:
```

```
    phone3=""
email=email_var.get()
dob=dob_var.get()
group=group_var.get()
address =address_var.get()
profession= profession_var.get()

#validating
if phone!="":
    if not(phone.isdigit()):
        mb.showerror("Number can contain only digits","Number can contain only digits")
        r.destroy()
        c=1
if email!="":
    if not('@' in email):
        mb.showerror("Invalid Email","Invalid Email")
        r.destroy()
        c=1

if phone2 !="":
    if not(phone2.isdigit()):
        mb.showerror("Number can contain only digits","Number can contain only digits")
        r.destroy()
        c=1

if phone3 !="":
    if not(phone3.isdigit()):
        mb.showerror("Number can contain only digits","Number can contain only digits")
        r.destroy()
```

```
            c=1

        if dob!="":
            if len(dob)!=10 and not("/" in dob):
                mb.showerror("Invalid Date of Birth Format","Invalid Date of Birth Format")
                r.destroy()
                c=1
        if profession!="":
            if not profession.isalpha():
                mb.showerror("Profession Can't contain digits or special characters","Profession Can't
contain digits or special characters")
                r.d estroy()
                c=1

        if c==0:

phonebook.append([name,phone,phone2,phone3,email,dob,group,address,profession,unique_id])
            mb.showinfo("Contact Created", "The Contact is Successfully Created")
            r.destroy()

        name_var.set("")
        phone_var.set("")
        email_var.set("")
        dob_var.set("")
        group_var.set("")
        address_var.set("")
        profession_var.set("")

        print(phonebook)
```

```
def add2ndnum():
    nonlocal phone2_var

    phone2_label=Label(r,
    text="Number 2:",
    font =("Times New Roman",15),
    bg="#c3e8e2",
    fg="BLACK",
    ).pack(side=TOP,anchor=N,fill=X,padx=100)
    phone2_entry=Entry(r,width=60,textvariable=phone2_var,).pack(side=TOP,anchor=N,padx=100)

    add=Button(r,
    text="ADD THIRD NUMBER",
    command=add3rdnum,
    font=("Times New Roman",15,"bold"),
    bg="#9ccefa",
    fg="black",).pack(side=TOP,anchor=N,fill=X,padx=200)
```

```
def add3rdnum():
    nonlocal phone3_var

    phone3_label=Label(r,
    text="Number 3:",
    font =("Times New Roman", 15),
    bg="#c3e8e2",
    fg="black",
    ).pack(side=TOP,anchor=N,fill=X,padx=100)
    phone3_entry=Entry(r,width=60,textvariable=phone3_var,).pack(side=TOP,anchor=N,padx=100)




phone2_var.set("")
phone3_var.set("")

sub=Button(r,
    text="Submit",
    command=submit,
    bg="#9ccefa",
    font=("Times New Roman",15,"bold"),
    pady=8,
    fg="black",
    relief=GROOVE,
    )
    sub.pack(side=TOP,anchor=N, fill=X,padx=200,pady=9)

    name_label=Label(r,
    text="Name:",
    font =("Times New Roman",15),
    bg="#c3e8e2",
    fg="black",
    ).pack(side=TOP,anchor=N,fill=X,padx=100)
    name_entry=Entry(r,width=100,textvariable=name_var,font = ('Times New
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100)

    phone_label=Label(r,
    text="Number 1:",
    font =("Times New Roman",15),
    bg="#c3e8e2",
    fg="black",
    ).pack(side=TOP,anchor=N,fill=X,padx=100)
    phone_entry=Entry(r,width=60,textvariable=phone_var,font = ('Times New
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100)
    add=Button(r,
        text="ADD SECOND NUMBER",
        command=add2ndnum,
        font=("Times New Roman",15,"bold"),
        bg="#9ccefa",
        fg="black",).pack(side=TOP,anchor=N,fill=X,padx=200)

    email_label=Label(r,text="Email:",
        font =("Times New Roman",15),
        bg="#c3e8e2",
        fg="black",
        ).pack(side=TOP,anchor=N,fill=X,padx=100)
    email_entry=Entry(r,width=100,textvariable=email_var,font = ('Times New
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100)

    dob_label=Label(r,text="Date of birth(DD/MM/YYYY):",
        font =("Times New Roman",15),
        bg="#c3e8e2",
        fg="black",
        ).pack(side=TOP,anchor=N,fill=X,padx=100)
    dob_entry=Entry(r,width=50,textvariable=dob_var,font = ('Times New
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100)



category_label=Label(r,text="Category(Family/Friends/Work/Favorite/Emergency/Important/Others):"
,
        font =("Times New Roman",15),
        bg="#c3e8e2",
        fg="black",
        ).pack(side=TOP,anchor=N,fill=X,padx=100)
    category_entry=Entry(r,width=60,textvariable=group_var,font = ('Times New
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100)


    address_label=Label(r,text="Address:",
        font =("Times New Roman",15),
        bg="#c3e8e2",
        fg="black",
        ).pack(side=TOP,anchor=N,fill=X,padx=100)
    address_entry=Entry(r,width=60,textvariable=address_var,font = ('Times New
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100)

    profession_label=Label(r,text="Designation:",
        font =("Times New Roman",15),
        bg="#c3e8e2",
        fg="black",
        ).pack(side=TOP,anchor=N,fill=X,padx=100)
    profession_entry=Entry(r,width=60,textvariable=profession_var,font = ('Times New
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100)



    r.mainloop()

def editContact(i,t,t1):
    r= object("Edit Contact", "1100x600",(1100,600),"#CED9F7")
    Title= Label(r,
    text="Edit Contact", relief=GROOVE,bg="#CBF7F3",fg="black",font=("Times New
Roman",30,"bold"),borderwidth=10
    ).pack(side=TOP,fill=X)
    i[1]=str(i[1])
    i[2]=str(i[2])
    i[3]=str(i[3])
    name_var=StringVar(master=r)
    phone_var=StringVar(master=r)
    phone2_var=StringVar(master=r)
    phone3_var=StringVar(r)
```

```
email_var=StringVar(r)
dob_var=StringVar(r)
group_var=StringVar(r)
profession_var=StringVar(r)
address_var=StringVar(r)

def submit():
    c=0
    unique_id = i[9]
    name=name_var.get()
    phone=phone_var.get()

    if phone2_var!=NoneType:
        phone2=phone2_var.get()
        phone2_var.set("")
    else:
        phone2=""
    if phone3_var!=NoneType:
        phone3=phone3_var.get()

        phone3_var.set("")
    else:
        phone3=""
    email=email_var.get()
    dob=dob_var.get()
    group=group_var.get()
    address =address_var.get()
    profession= profession_var.get()
    d=phonebook.index(i)
```

```
#validating
if phone!="":
    if not(phone.isdigit()):
        mb.showerror("Number can contain only digits","Number can contain only digits")
        r.destroy()
        c=1
if email!="":
    if not('@' in email):
        mb.showerror("Invalid Email","Invalid Email")
        r.destroy()
        c=1

if phone2 !="":
    if not(phone2.isdigit()):
        mb.showerror("Number can contain only digits","Number can contain only digits")
        r.destroy()
        c=1

if phone3 !="":
    if not(phone3.isdigit()):
        mb.showerror("Number can contain only digits","Number can contain only digits")
        r.destroy()
        c=1

if dob!="":
    if len(dob)!=10 and not("/" in dob):
        mb.showerror("Invalid Date of Birth Format","Invalid Date of Birth Format")
        r.destroy()
        c=1
        if profession!="":
            if not profession.isalpha():
                mb.showerror("Profession Can't contain digits or special characters","Profession Can't contain digits or special characters")
                r.destroy()
                c=1

        if c==0:
            phonebook[d]=[name,phone,phone2,phone3,email,dob,group,address,profession,unique_id]
            print(phonebook)

            mb.showinfo("Contact Updated", "The Contact is Successfully Updated")
            r.destroy()
        t.destroy()
        t1.destroy()




    sub=Button(r,
        text="Update Contact",
        command=submit,
        bg="#9ccefa",
        font=("Times New Roman",15,"bold"),
        pady=8,
        fg="black",
        relief=GROOVE,).pack(side=TOP,anchor=N,padx=200,pady=9)

    name_var.set(i[0])
    phone_var.set(i[1])
    phone2_var.set(i[2])
    phone3_var.set(i[3])
    email_var.set(i[4])
    dob_var.set(i[5])
    group_var.set(i[6])
    address_var.set(i[7])
    profession_var.set(i[8])

    name_label=Label(r,
    text="Name:",
    font =("Times New Roman",15),
    bg="#c3e8e2",
    fg="black",
    ).pack(side=TOP,anchor=N,fill=X,padx=100)
    name_entry=Entry(r,width=100,textvariable=name_var,font = ('Times New
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100)
    phone_label=Label(r,
    text="Number 1:",
    font =("Times New Roman",15),
    bg="#c3e8e2",
    fg="black",
    ).pack(side=TOP,anchor=N,fill=X,padx=100)
    phone_entry=Entry(r,width=60,textvariable=phone_var,font = ('Times New
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100)
    phone2_label=Label(r,
        text="Number 2:",
        font =("Times New Roman",15),
        bg="#c3e8e2",
        fg="black",
        ).pack(side=TOP,anchor=N,fill=X,padx=100)
    phone2_entry=Entry(r,width=60,textvariable=phone2_var,).pack(side=TOP,anchor=N,padx=100)
    phone3_label=Label(r,
        text="Number 3:",
        font =("Times New Roman",15),
        bg="#c3e8e2",
        fg="black",
        ).pack(side=TOP,anchor=N,fill=X,padx=100)
    phone3_entry=Entry(r,width=60,textvariable=phone3_var,).pack(side=TOP,anchor=N,padx=100)
    email_label=Label(r,text="Email:",
        font =("Times New Roman",15),
        bg="#c3e8e2",
        fg="black",
        ).pack(side=TOP,anchor=N,fill=X,padx=100)
    email_entry=Entry(r,width=100,textvariable=email_var,font = ('Times New
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100)
    dob_label=Label(r,text="Date of birth(DD/MM/YYYY):",
        font =("Times New Roman",15),
        bg="#c3e8e2",
        fg="black",
        ).pack(side=TOP,anchor=N,fill=X,padx=100)
    dob_entry=Entry(r,width=60,textvariable=dob_var,font = ('Times New
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100)

category_label=Label(r,text="Category(Family/Friends/Work/Favorite/Emergency/Important/Others):"
,
        font =("Times New Roman",15),
        bg="#c3e8e2",
        fg="black",
        ).pack(side=TOP,anchor=N,fill=X,padx=100)
    category_entry=Entry(r,width=60,textvariable=group_var,font = ('Times New 
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100)

    address_label=Label(r,text="Address:",
        font =("Times New Roman",15),
        bg="#c3e8e2",
        fg="black",
        ).pack(side=TOP,anchor=N,fill=X,padx=100)
    address_entry=Entry(r,width=100,textvariable=address_var,font = ('Times New 
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100)
    profession_label=Label(r,text="Designation :",
        font =("Times New Roman",15),
        bg="#c3e8e2",
        fg="black",
        ).pack(side=TOP,anchor=N,fill=X,padx=100)
    profession_entry=Entry(r,width=60,textvariable=profession_var,font = ('Times New 
Roman',10,'normal')).pack(side=TOP,anchor=N,padx=100)

    r.mainloop()


def delete(r1,r,i):
        #i=phonebook.index(i)
        r2=object( "{} Contact Deleted".format(i[0]), "300x100",(300,100),"#CED9F7")
        r2.maxsize(300,100)
        i[1]=str(i[1])
        i[2]=str(i[2])
        i[3]=str(i[3])
        phonebook.remove(i)

        Label(r2,text="Contact Deleted",padx=10,pady=10,bg="#c3e8e2",
        font=("Times New Roman",15),
        fg="black",relief=GROOVE).pack(side=TOP,anchor=CENTER,fill=X)

        r1.destroy()
        r.destroy()

        Button(
            r2,
            text="OK",
            font=("Times New Roman",15,"bold"),
            bg="#9ccefa",
            fg="black",
            command=r2.destroy
            ).pack(side=BOTTOM,anchor=S,fill=X,padx=10,pady=10)



        r2.mainloop()

def open(i,r):
    r1=object( "{} Contact Info".format(i[0]), "500x500",(500,500),"#CED9F7")
    Title= Label(r1,
    text="Contact Details", relief=GROOVE,bg="#CBF7F3",fg="black",font=("Times New
Roman",30,"bold"),borderwidth=10
        ).pack(side=TOP,fill=X)
```

```
Label(r1,
   text="Name: "+ i[0],
   font =("Times New Roman",15),
   bg="#b8cde0",
   fg="black",
   relief=SUNKEN,
   borderwidth=2,
   padx=10,
      pady=3).pack(fill=X,side=TOP)

Label(r1,
      text="Number: " +str(i[1]),
      font =("Times New Roman",15),
      bg="#b8cde0",
      fg="black",
      relief=SUNKEN,
      borderwidth=2,
      padx=10,
      pady=3,
      ).pack(fill=X,side=TOP)

if str(i[2])!="":
   Label(r1,
         text="Number 2: " +str(i[2]),
         font =("Times New Roman",15),
         bg="#b8cde0",
         fg="black",
         relief=SUNKEN,
            borderwidth=2,
            padx=10,
            pady=3,
            ).pack(fill=X,side=TOP)

if str(i[3])!="":
    Label(r1,
            text="Number 3: " +str(i[3]),
            font =("Times New Roman",15),
            bg="#b8cde0",
            fg="black",
            relief=SUNKEN,
            borderwidth=2,
            padx=10,
            pady=3,
            ).pack(fill=X,side=TOP)

Label(r1,
        text="Email: " +str(i[4]),
        font =("Times New Roman",15),
        bg="#b8cde0",
        fg="black",
        relief=SUNKEN,
        borderwidth=2,
        padx=10,
        pady=3,
        ).pack(fill=X,side=TOP)

Label(r1,
        text="Date of Birth: " +str(i[5]),
        font =("Times New Roman",15),
        bg="#b8cde0",
        fg="black",
        relief=SUNKEN,
        borderwidth=2,
        padx=10,
        pady=3,
        ).pack(fill=X,side=TOP)


Label(r1,
        text="Group: " +i[6],
        font =("Times New Roman",15),
        bg="#b8cde0",
        fg="black",
        relief=SUNKEN,
        borderwidth=2,
        padx=10,
        pady=3,
        ).pack(fill=X,side=TOP)

Label(r1,
        text="Address: " +str(i[7]),
        font =("Times New Roman",15),
        bg="#b8cde0",
        fg="black",
        relief=SUNKEN,
        borderwidth=2,
        padx=10,
        pady=3,
        ).pack(fill=X,side=TOP)

Label(r1,
        text="Designation: " +str(i[8]),
        font =("Times New Roman",15),
        bg="#b8cde0",
        fg="black",
        relief=SUNKEN,
        borderwidth=2,
        padx=10,
        pady=3,
        ).pack(fill=X,side=TOP)

Label(r1,
        text="Timestamp: " +str(i[9]),
        font =("Times New Roman",15),
        bg="#b8cde0",
        fg="black",
        relief=SUNKEN,
        borderwidth=2,
        padx=10,
        pady=3,
        ).pack(fill=X,side=TOP)


"""
edit_img=PhotoImage(r1,file="D:\phone.py\icons8-edit-40.png")
```

```
p6 = edit_img.subsample(1,1)

delete1_img=PhotoImage(r1,file="D:\phone.py\icons8-delete-300.png")
p7 = delete1_img.subsample(6,6)
"""
g_img=PhotoImage(master=r1,file="D:\phone.py\icons8-delete-300.png")
i1 = g_img.subsample(6,6)

g1_img=PhotoImage(master=r1,file="D:\phone.py\icons8-edit-80.png")
i2 = g1_img.subsample(2,2)

Button(r1,
        text=" Edit Contact",
        command=lambda:[editContact(i,r,r1)],
        font=("Times New Roman",15),
        fg="black",
        bg="#9ccefa",
        pady=5,
        image=i2,
        compound=LEFT,
        ).pack(side=BOTTOM,fill=X)



Button(r1,text= " Delete Contact",
        #way to send parameters
        #command=lambda:delete(r1),
        command=lambda:[delete(r1,r,i)],
        font=("Times New Roman",15),
        fg="black",
        bg="#9ccefa",
        pady=5,
        #image=p7,
        #compound=LEFT,
        image=i1,
        compound=LEFT,
        ).pack(side=BOTTOM,fill=X)


    r1.mainloop()

def openList(l):
    r=object("Contacts","1100x600",(1100,600),"#CED9F7")

    Title= Label(r,
    text="Contacts", relief=GROOVE,bg="#CBF7F3",fg="black",font=("Times New
Roman",30,"bold"),borderwidth=10
    ).pack(side=TOP,fill=X)

    create_img=PhotoImage(master=r,file="D:\phone.py\icons8-add-96.png")
    p = create_img.subsample(2,2)

    create_contact=Button(r,
    text=" Create Contact",
    image=p,
    font=("Times New Roman",14,"bold"),
    command=createContact,
    compound=LEFT,
    bg="#9ccefa",
fg="black"
).pack(padx=200,fill=X,pady=2)

search_var=StringVar(r)

def submit():
    global phonebook
    q=search_var.get()
    print(q,type(q))
    l=[]
    c=0
    for i in phonebook:
        print(i)

        if i[0].startswith(q) or i[0].endswith(q) or i[1]==q or i[2]==q or i[3]==q or i[4]==q or i[6]==q or
i[7].startswith(q) or i[7].endswith(q) or i[8]==q:
                l.append(i)
                c=1

        print("c=",c)
    if c==0:
        mb.showinfo("No Contact Found","No results found")
    elif c==1:
        r.destroy()
        openList(l)



search_label=Label(r,text="Search :",
    font =("Times New Roman",15,"bold"),
    bg="#CBF7F3",
    fg="black",
    ).pack(side=TOP,anchor=N,fill=X,pady=4)
search_entry=Entry(r,textvariable=search_var,font = ('Times New
Roman',10,'normal'),width=100).pack(side=TOP,anchor=N)

g1_img=PhotoImage(master=r,file="D:\phone.py\icons8-search-96.png")
i2 = g1_img.subsample(3,3)

Button(r,
    text=" Search",
    command=submit,
    bg="#9ccefa",
    fg="black",
    font=("Times New Roman",15,"bold"),
    padx=10,
    pady=5,
    image=i2,
    compound=LEFT
    ).pack(side=TOP,anchor=N,padx=200,fill=X,pady=2)



#phonebook.append([name,phone,phone2,phone3,email,dob,group,address,profession,unique_id])

scroll_bar = Scrollbar(r)

scroll_bar.pack( side = RIGHT,
        fill = Y )

columns = ('name', 'phone_number_1',"phone_number_2","phone_number_3",
'email',"date_of_birth","group","address","profession","unique_id")
tree = ttk.Treeview(r, columns=columns, show='headings',yscrollcommand = scroll_bar.set)
tree.heading("name",text="Name")
tree.heading("phone_number_1",text="Phone Number 1")
tree.heading("phone_number_2",text="Phone Number 2")
tree.heading("phone_number_3",text="Phone Number 3")
tree.heading("email",text="Email")
tree.heading("date_of_birth",text="Date of Birth")
tree.heading("group",text="Group")
tree.heading("address",text="Address")
tree.heading("profession",text="Profession")
tree.heading("unique_id",text="Time Stamp")

tree.column('name', width=60, anchor=W)
tree.column('phone_number_1', width=15,)
tree.column('phone_number_2', width=15,)
tree.column('phone_number_3', width=15, )
tree.column('email', width=60, )
tree.column('date_of_birth',width=8,)
tree.column('group', width=20, )
tree.column('address', width=50, )
tree.column('profession', width=10,)
tree.column('unique_id', width=10,)

l.sort(key=lambda x:x[0])
for i in l:
    tree.insert('', END, values=i)

def item_selected(event):
    for selected_item in tree.selection():
        item = tree.item(selected_item)
        record = item['values']
        print(record)
        open(record,r)

tree.bind('<Double-1>', item_selected)

tree.pack(fill=BOTH,expand=True)

#scrollbar = ttk.Scrollbar(r, orient=VERTICAL, command=tree.yview)
#tree.configure(yscroll=scrollbar.set)
#scrollbar.pack(side=LEFT,fill=Y)

r.mainloop()

def dummyData():
#phonebook.append([name,phone,phone2,phone3,email,dob,group,address,profession,unique_id])
#Dummy data is to add entries quickly so that testing can be done quickly
phonebook.append(["Police","100","","","gov@gmail.com","","emergency","Ranchi
Jharkhand","police",str(datetime.datetime.now())])
phonebook.append(["Hospital","112","","","hospital@gmail.com","","emergency","Ranchi
Jharkhand","hospital",str(datetime.datetime.now())])
```

```
    phonebook.append(["Shreya Mishra","9193939999","9343387733","9876543210","shreya@gmail.com","22/02/2002","friend","Ranchi Jharkhand","student",str(datetime.datetime.now())])
    phonebook.append(["Apoorva Raj","9193933499","9343387732","9126543210","apoorva@gmail.com","25/12/2002","important","Mysore Karnataka","student",str(datetime.datetime.now())])
    phonebook.append(["Adrija Pes","9419393559","9343456733","9876733210","adrija@gmail.com","05/05/2004","favourite","Bangalore Karnataka","student",str(datetime.datetime.now())])
    phonebook.append(["Abhijeet Mishra","9193456999","9999387733","9873293210","abhijeet@gmail.com","09/11/1978","work","Hyderabad Telangana","teacher",str(datetime.datetime.now())])
    phonebook.append(["Isham Sinha","9567456999","9934587733","9878943210","isham@gmail.com","09/01/2007","friend","Bangalore Karnataka","student",str(datetime.datetime.now())])
    phonebook.append(["B. S. Grewal","9193456999","9999387733","9873293210","grewal@gmail.com","07/08/2002","classmate","Hyderabad Telangana","teacher",str(datetime.datetime.now())])
    phonebook.append(["Skanda Prasad","9193402399","9991234733","9870912210","skanda@gmail.com","04/02/2005","friend","Bangalore Karnataka","student",str(datetime.datetime.now())])
    phonebook.append(["Chandan Kumar Jha","9155402399","9991111133","9222912210","chandan@gmail.com","04/12/1985","family","Bangalore Karnataka","student",str(datetime.datetime.now())])
    phonebook.append(["Rashmi Jha","9155433399","9333111133","9233312210","rashmi@gmail.com","02/07/1985","family","Bangalore Karnataka","student",str(datetime.datetime.now())])
    mb.showinfo("Data added", "Sample Data Entries added to the phone directory.")
```

```
def searchContact():
    global phonebook

    r= object("Search Contact", "600x600",(600,600),"#CED9F7")
    Title= Label(r,
    text="Search Contact", relief=GROOVE,bg="#CBF7F3",fg="black",font=("Times New
Roman",30,"bold"),borderwidth=10
    ).pack(side=TOP,fill=X)

    search_var=StringVar(r)

    def submit():
        global phonebook
        q=search_var.get()
        print(q,type(q))
        l=[]
        c=0
        for i in phonebook:
            print(i)

            if i[0].startswith(q) or i[0].endswith(q) or i[1]==q or i[2]==q or i[3]==q or i[4]==q or i[6]==q or
            i[7].startswith(q) or i[7].endswith(q) or i[8]==q:
                    l.append(i)
                    c=1

            print("c=",c)
        if c==0:
            mb.showinfo("No Contact Found","No results found")
        elif c==1:
```

```
        openList(l)


    search_label=Label(r,text="Search:",
        font =("Times New Roman",15,"bold"),
        bg="#b8cde0",
        fg="black",
        ).pack(side=TOP,anchor=N,fill=X,pady=4)
    search_entry=Entry(r,textvariable=search_var,font = ('Times New
Roman',10,'normal'),width=100).pack(side=TOP,anchor=N)

    g1_img=PhotoImage(master=r,file="D:\phone.py\icons8-search-96.png")
    i2 = g1_img.subsample(3,3)

    Button(r,
        text="Search",
        command=submit,
        bg="#9ccefa",
        fg="black",
        font=("Times New Roman",15,"bold"),
        padx=10,
        pady=5,
        image=i2,
        compound=LEFT
        ).pack(side=TOP,anchor=N,padx=200,fill=X,pady=10)


    r.mainloop()
```

```
def allContacts():
    openList(phonebook)

    #phonebook.append([name,phone,phone2,phone3,email,dob,group,address,profession,id])

def manageGroups():
    r=object("Groups","600x600",(600,600),"#CED9F7")
    Title= Label(r,
    text="Groups", relief=GROOVE,bg="#CBF7F3",fg="black",font=("Times New
Roman",30,"bold"),borderwidth=10
    ).pack(side=TOP,fill=X)
    def exist():
        t=["work","friends","friend","family","emergency","favourite","important"]
        l=[]
        for i in phonebook:
            if i[6].lower() not in t:
                l.append(i[6])

        return l

    def work():
        l=[]
        for i in phonebook:
            if i[6]=="Work" or i[6]=="WORK" or i[6]=="work":
                l.append(i)
        if l!=[]:
            openList(l)
        else:
            mb.showinfo("Empty Group","No one present in the group")
```

```
 def friend():
    l=[]
    for i in phonebook:
       if i[6]=="Friends" or i[6]=="FRIENDS" or i[6]=="friends" or i[6]=="friend" or i[6]=="Friend"
or i[6]=="FRIEND":
          l.append(i)
    if l!=[]:
       openList(l)
    else:
       mb.showinfo("Empty Group","No one present in the group")

 def family():
    l=[]
    for i in phonebook:
       if i[6]=="Family" or i[6]=="FAMILY" or i[6]=="family":
          l.append(i)
    if l!=[]:
       openList(l)
    else:
       mb.showinfo("Empty Group","No one present in the group")

 def favourite():
    l=[]
    for i in phonebook:
       if i[6]=="favourite" or i[6]=="Favourite" or i[6]=="favourite".upper():
          l.append(i)
    if l!=[]:
       openList(l)
    else:
        mb.showinfo("Empty Group","No one present in the group")

def important():
    l=[]
    for i in phonebook:
        if i[6]=="important" or i[6]=="Important" or i[6]=="important".upper():
            l.append(i)
    if l!=[]:
        openList(l)
    else:
        mb.showinfo("Empty Group","No one present in the group")

def emergency():
    l=[]
    for i in phonebook:
        if i[6]=="emergency" or i[6]=="Emergency" or i[6]=="EMERGENCY":
            l.append(i)
    if l!=[]:
        openList(l)
    else:
        mb.showinfo("Empty Group","No one present in the group")

def others():
    l=[]
    a=exist()
    for i in phonebook:
        if i[6] in a:
            l.append(i)
    if l!=[]:
        openList(l)
    else:
        mb.showinfo("Empty Group","No one present in the group")

g_img=PhotoImage(master=r,file="D:\phone.py\icons8-emergency-64.png")
i = g_img.subsample(2,2)

g1_img=PhotoImage(master=r,file="D:\phone.py\icons8-work-64.png")
i1 = g1_img.subsample(2,2)

g2_img=PhotoImage(master=r,file="D:\phone.py\icons8-friends-96.png")
i2 = g2_img.subsample(2,2)

g3_img=PhotoImage(master=r,file="D:\phone.py\icons8-family-120.png")
i3 = g3_img.subsample(3,3)

g4_img=PhotoImage(master=r,file="D:\phone.py\icons8-group-96 (1).png")
i4 = g4_img.subsample(2,2)

g5_img=PhotoImage(master=r,file="D:\phone.py\icons8-favourite-64.png")
i5 = g5_img.subsample(2,2)

g6_img=PhotoImage(master=r,file="D:\phone.py\icons8-important-64.png")
i6 = g6_img.subsample(2,2)
```

```
Button(
   r,
   text=" Emergency Contacts",
   command=emergency,
   bg="#9ccefa",
   fg="black",
   font =("Times New Roman",15),
   compound=LEFT,
   image=i

   ).pack(fill=X,pady=5)

Button(
   r,
   text=" Important Contacts",
   command=important,
   bg="#9ccefa",
   fg="black",
   font =("Times New Roman",15),
   compound=LEFT,
   image=i6

   ).pack(fill=X,pady=5)

Button(
   r,
   text=" Favourite Contacts",
   command=favourite,
   bg="#9ccefa",
    fg="black",
    font =("Times New Roman",15),
    compound=LEFT,
    image=i5

    ).pack(fill=X,pady=5)

Button(
    r,
    text=" Work",
    command=work,
    bg="#9ccefa",
    fg="black",
    font =("Times New Roman",15),
    compound=LEFT,
    image=i1

    ).pack(fill=X,pady=5)

Button(
    r,
    text=" Friends",
    command=friend,
    bg="#9ccefa",
    fg="black",
    font =("Times New Roman",15),
    compound=LEFT,
    image=i2
    ).pack(fill=X,pady=5)
    Button(
        r,
        text=" Family",
        command=family,
        bg="#9ccefa",
        fg="black",
        font =("Times New Roman",15),
        compound=LEFT,
        image=i3
        ).pack(fill=X,pady=5)
    Button(
        r,
        text=" Others",
        command=others,
        bg="#9ccefa",
        fg="black",
        font =("Times New Roman",15),
        compound=LEFT,
        image=i4
        ).pack(fill=X,pady=5)

    r.mainloop()

def deleteallContact():
    global phonebook
    phonebook=[]
    print(phonebook)
    mb.showinfo("All contacts Deleted","All Contacts Deleted")
```

```
def exit_prg():
    sys.exit(0)

def setPhonebook(pb):
    global phonebook
    phonebook =pb

def dark_mode():
    import function
    function.setPhonebook(phonebook)
    import main
```

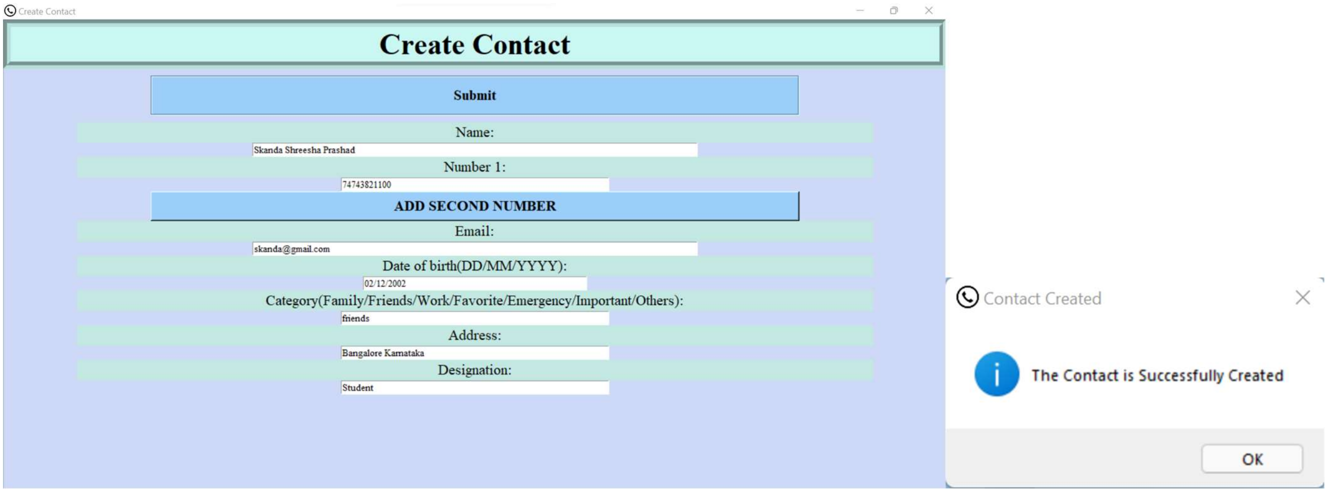
TESTING

We have tested and made sure that there are no logical and syntax errors during the execution of the project. We have added two backgrounds, a Light Mode and a Dark Mode for aesthetic purposes. The user can switch between them anytime. We have tested every function present in the application to make sure that they are all error free. To prevent a person from entering alphabets as the input for phone numbers, we have ensured that the new person's list will get appended only if the phone numbers contain numbers, and not alphabets and special characters.

Every time the user adds a person's details, the person's details are added to the directory and is then displayed when All Contacts is opened. The user can modify a person's details by clicking on their name in in the All-Contacts tab. A person's contact can also be deleted in the same place. The application also allows the user to delete all the contacts present in one go. If the user clicks on Delete All Contacts, every single contact and their details present in the application will be erased.

We have also enabled the user to sort the contacts into different pre-existing groups based on their need. The application provides multiple groups, namely, Family, Friends, Work, Emergency Contacts, etc. We have also added the feature to select a contact as an Important Contact or a Favourite Contact.

One of the main features of the application is its ability to search for a contact. It allows them to search for specific contacts based on any criteria, be it their name, phone number, email address, address, category, etc. All the contacts are stored in alphabetical order so that they can be easily accessed. Each contact also stores the timestamp at which they have been added. This serves as a unique ID for each person.



Create Contact

Create Contact

Submit

Name:
Skanda Shreesha Prashad

Number 1:
74743821100

ADD SECOND NUMBER

Email:
skanda@gmail.com

Date of birth(DD/MM/YYYY):
02/12/2002

Category(Family/Friends/Work/Favorite/Emergency/Important/Others):
friends

Address:
Bangalore Karnataka

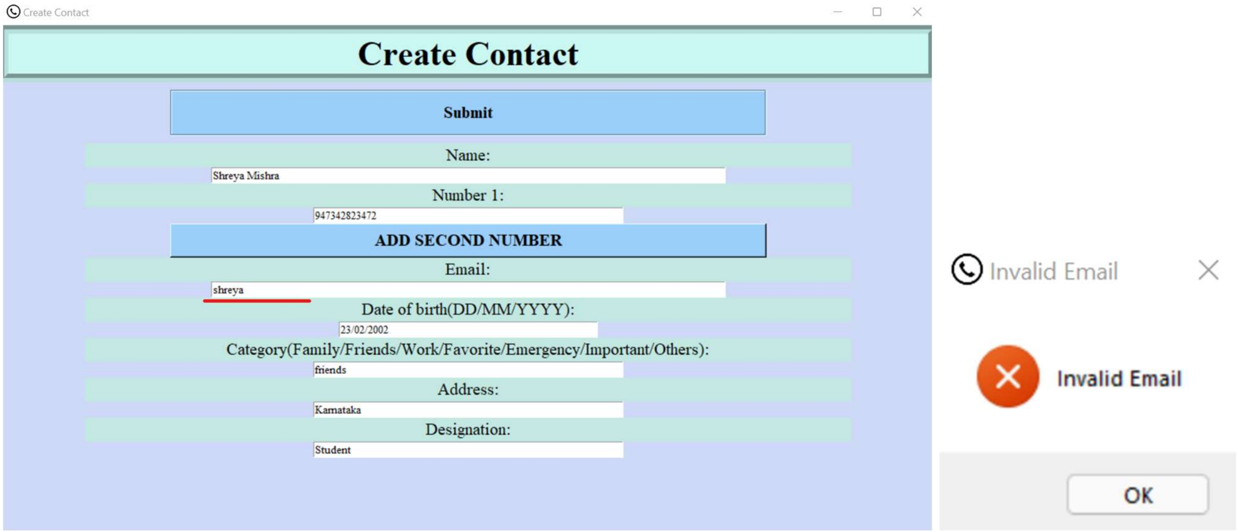
Designation:
Student

Contact Created

The Contact is Successfully Created

OK

In the another example, we put an invalid format of email



Create Contact

Create Contact

Submit

Name:
Shreya Mishra

Number 1:
947342823472

ADD SECOND NUMBER

Email:
shreya

Date of birth(DD/MM/YYYY):
23/02/2002

Category(Family/Friends/Work/Favorite/Emergency/Important/Others):
friends

Address:
Karnataka

Designation:
Student

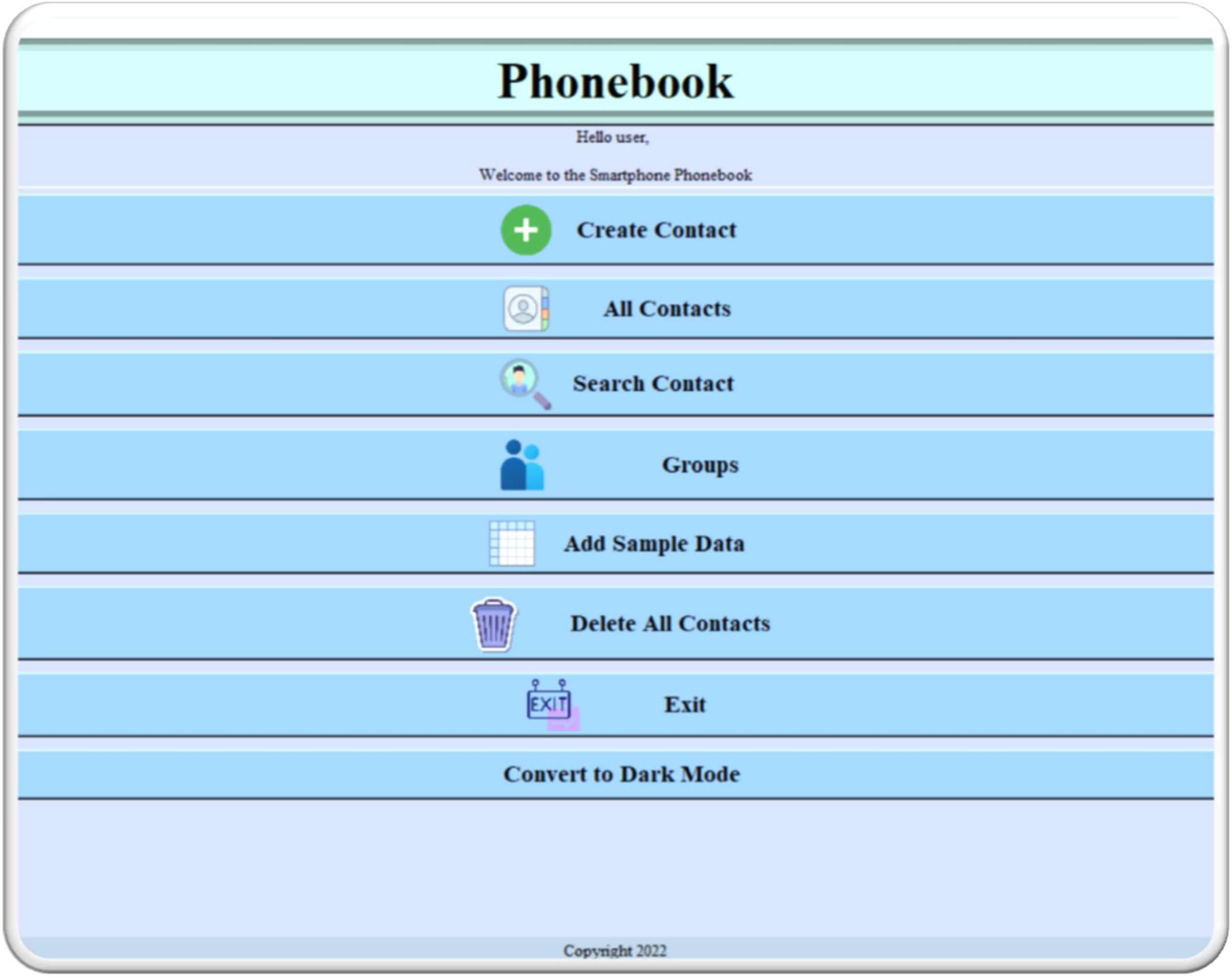
Invalid Email

Invalid Email

OK

Here , we get an error saying that the email is invalid. The same is true for numbers and Date of Birth format.

RESULT

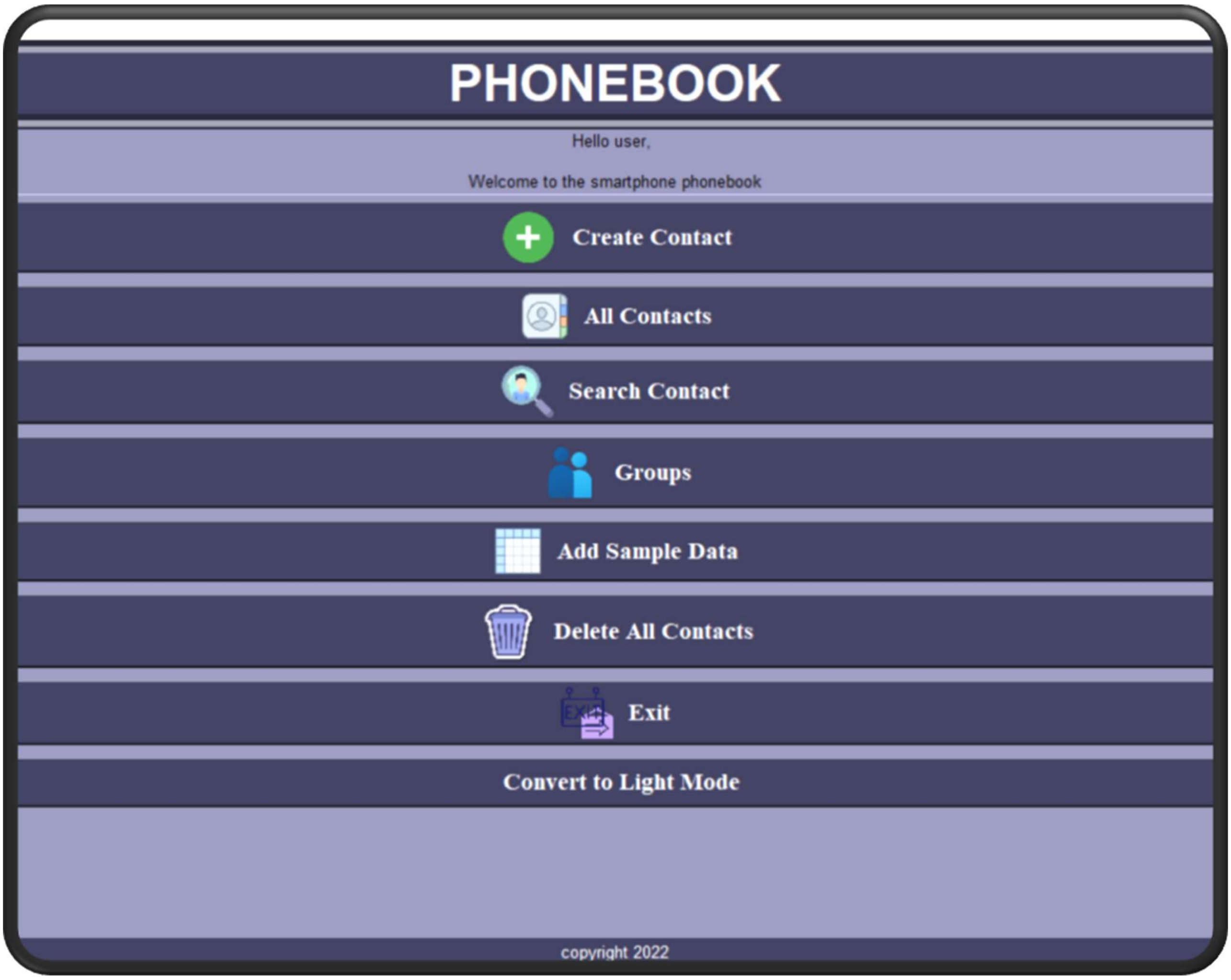


Phonebook

Hello user,

Welcome to the Smartphone Phonebook

The Menu Page in Light Mode



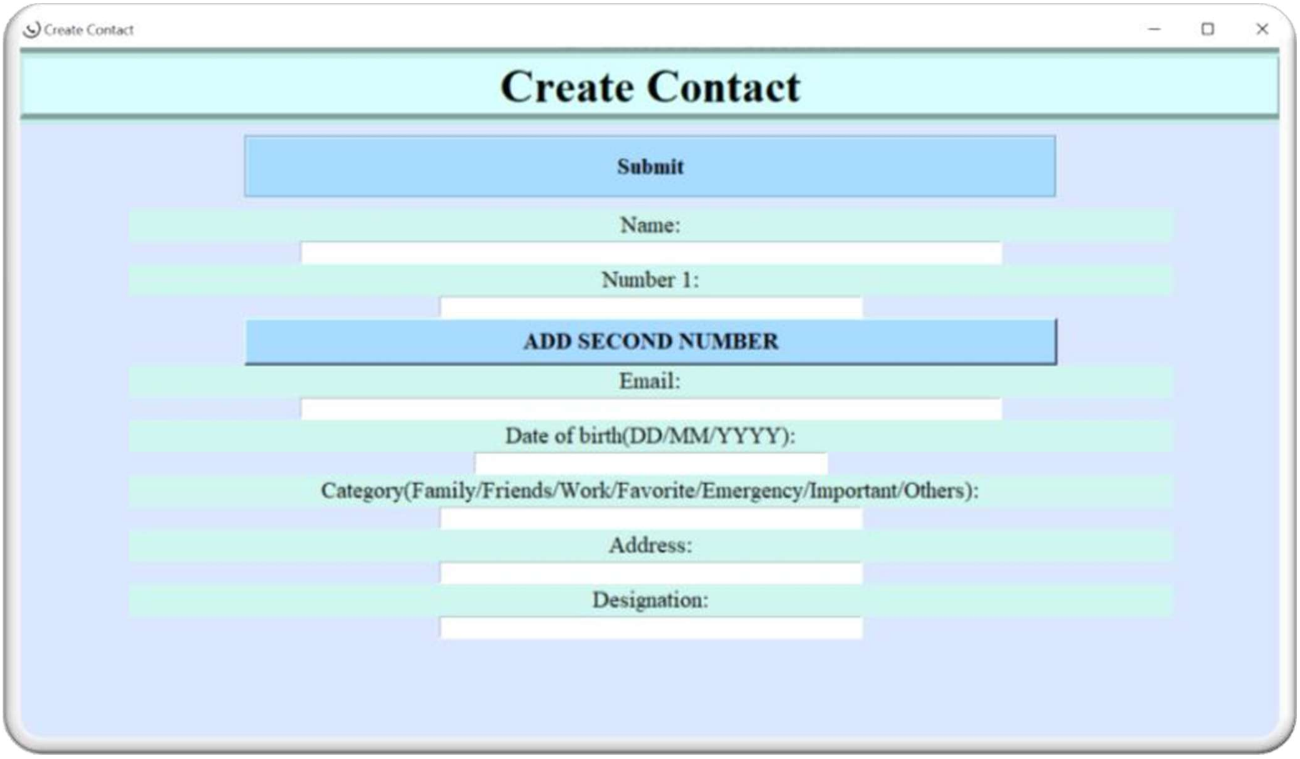
PHONEBOOK

Hello user,

Welcome to the smartphone phonebook

copyright 2022

The Menu Page in Dark Mode



Create Contact

Name:

Number 1:

Email:

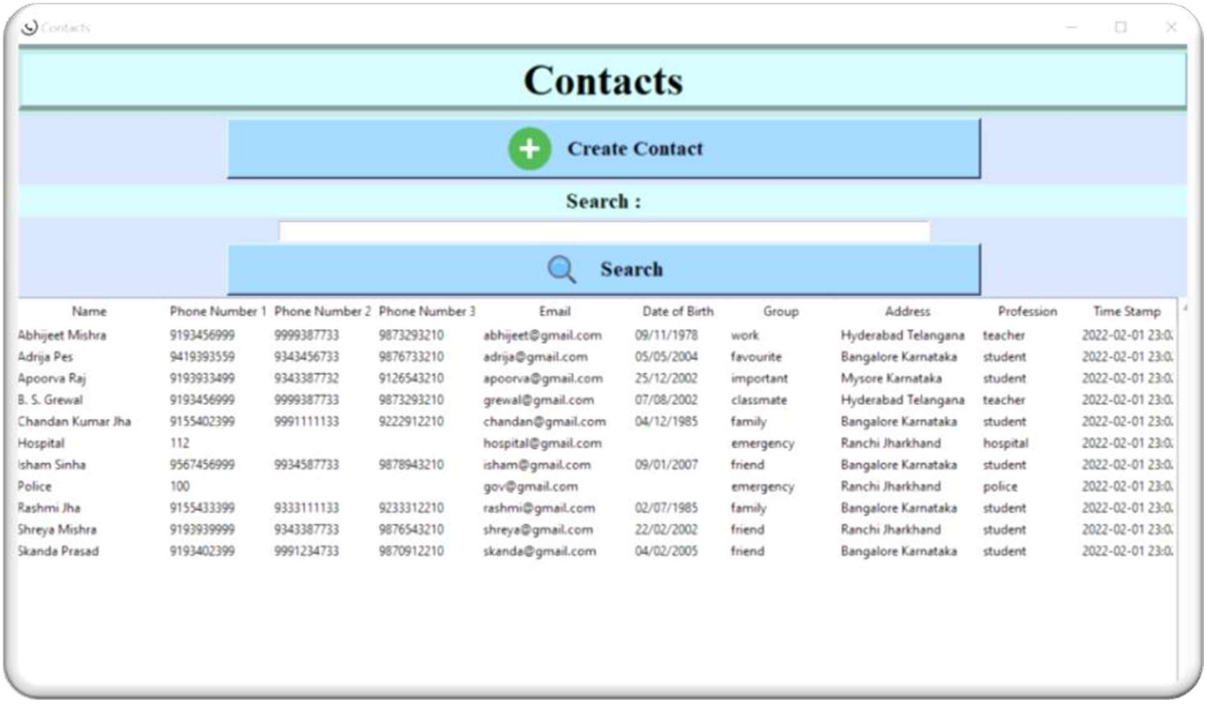
Date of birth(DD/MM/YYYY):

Category(Family/Friends/Work/Favorite/Emergency/Important/Others):

Address:

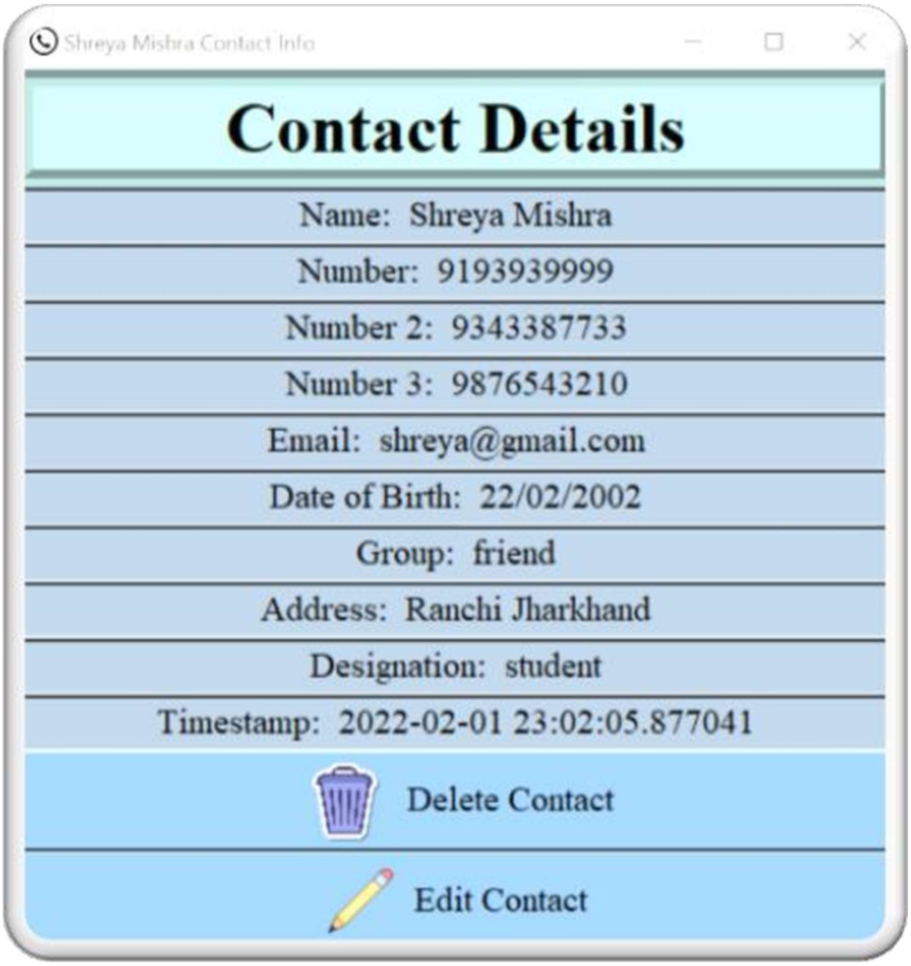
Designation:

Create Contact Page



Name	Phone Number 1	Phone Number 2	Phone Number 3	Email	Date of Birth	Group	Address	Profession	Time Stamp
Abhijeet Mishra	9193456999	9999387733	9873293210	abhijeet@gmail.com	09/11/1978	work	Hyderabad Telangana	teacher	2022-02-01 23:0
Adrija Pes	9419393559	9343456733	9876733210	adrija@gmail.com	05/05/2004	favourite	Bangalore Karnataka	student	2022-02-01 23:0
Apoorva Raj	9193933499	9343387732	9126543210	apoorva@gmail.com	25/12/2002	important	Mysore Karnataka	student	2022-02-01 23:0
B. S. Grewal	9193456999	9999387733	9873293210	grewal@gmail.com	07/08/2002	classmate	Hyderabad Telangana	teacher	2022-02-01 23:0
Chandan Kumar Jha	9155402399	9991111133	9222912210	chandan@gmail.com	04/12/1985	family	Bangalore Karnataka	student	2022-02-01 23:0
Hospital	112			hospital@gmail.com		emergency	Ranchi Jharkhand	hospital	2022-02-01 23:0
Isham Sinha	9567456999	9934587733	9878943210	isham@gmail.com	09/01/2007	friend	Bangalore Karnataka	student	2022-02-01 23:0
Police	100			gov@gmail.com		emergency	Ranchi Jharkhand	police	2022-02-01 23:0
Rashmi Jha	9155433399	9333111133	9233312210	rashmi@gmail.com	02/07/1985	family	Bangalore Karnataka	student	2022-02-01 23:0
Shreya Mishra	9193939999	9343387733	9876543210	shreya@gmail.com	22/02/2002	friend	Ranchi Jharkhand	student	2022-02-01 23:0
Skanda Prasad	9193402399	9991234733	9870912210	skanda@gmail.com	04/02/2005	friend	Bangalore Karnataka	student	2022-02-01 23:0

Contacts List



Shreya Mishra Contact Info

Contact Details

Name: Shreya Mishra

Number: 9193939999

Number 2: 9343387733

Number 3: 9876543210

Email: shreya@gmail.com

Date of Birth: 22/02/2002

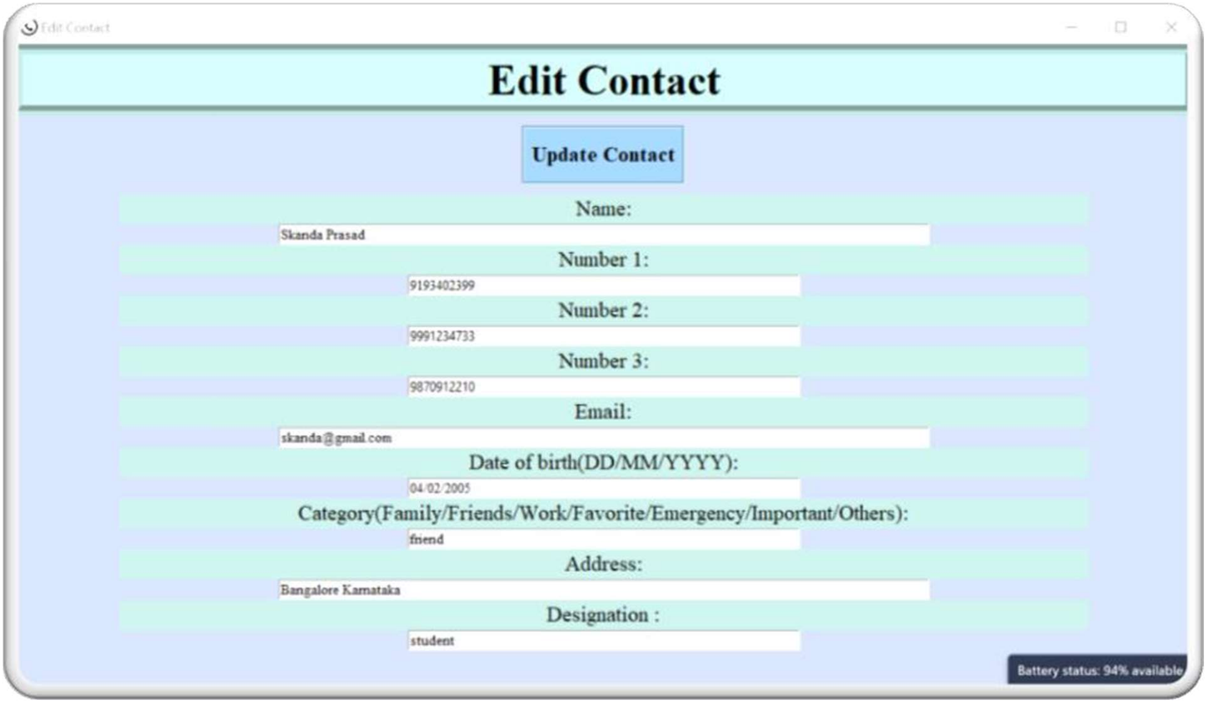
Group: friend

Address: Ranchi Jharkhand

Designation: student

Timestamp: 2022-02-01 23:02:05.877041

Individual Contact Details



Name:
Skanda Prasad

Number 1:
9193402399

Number 2:
9991234733

Number 3:
9870912210

Email:
skanda@gmail.com

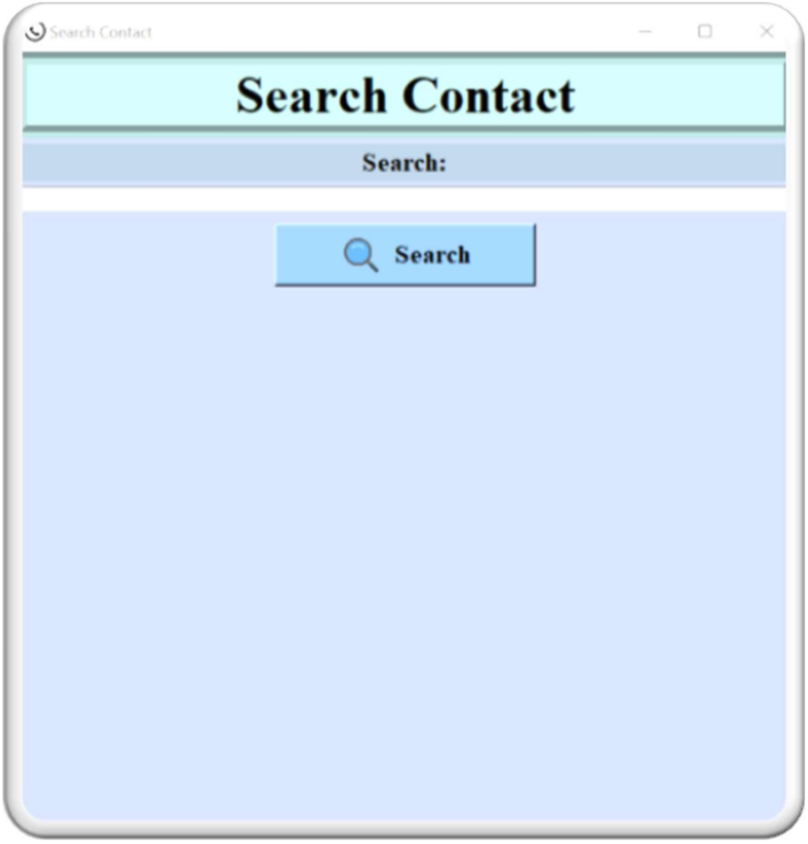
Date of birth(DD/MM/YYYY):
04/02/2005

Category(Family/Friends/Work/Favorite/Emergency/Important/Others):
friend

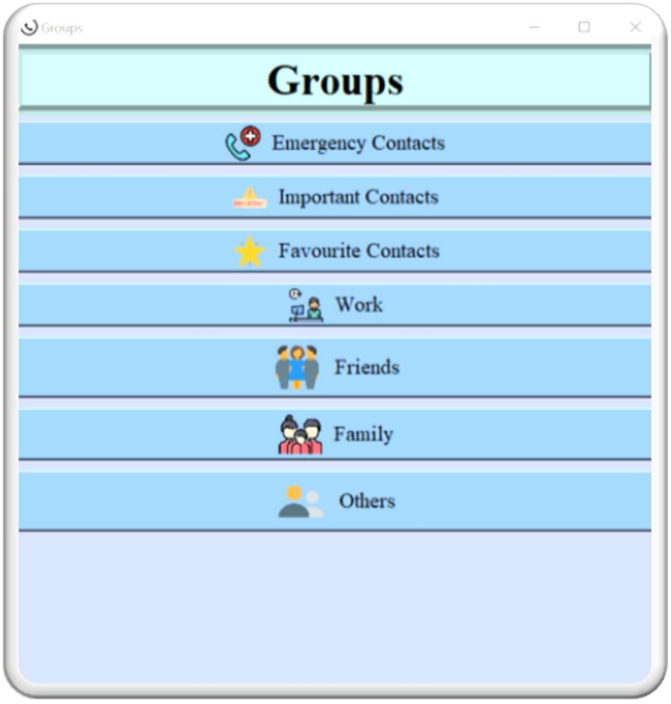
Address:
Bangalore Karnataka

Designation :
student

Edit Contact Page



Search Contact Page



Groups
Groups
Emergency Contacts
Important Contacts
Favourite Contacts
Work
Friends
Family
Others

Groups Page

ANALYSIS

The project has been tested to our abilities and we have not found any logical or syntax errors. Based on the reviews, the project has been updated to contain the unique ID generated using the timestamp, the ability to add multiple phone numbers, etc. Extra features such as the Groups, Dark to Light Mode conversion, and many other features have been added to the project. The project also has a Sample Data that has been included for ease of usage during demonstration and testing.

CONCLUSION

We thus conclude out project titled, ‘Phonebook”, made using Python and using environments such as PyCharm and Visual Studio Code. Phonebook is a user-friendly project that contains Graphical User Interface, which is convenient to a user who does not have any knowledge of coding. GUI has been implemented using Tkinter and many other modules have been utilized for various purposes such as the timestamp, etc. We have also given the user the choice to choose the mode in which they want the program to be displayed, i.e., Light Mode or Dark Mode.

FUTURE ENHANCEMENTS

No project is perfect as we are after all humans and humans are bound to make errors. We believe that our project could be enhanced in the following ways:

1. The Graphical User Interface could be modified and the colour scheme used could have been enhanced.
2. The ability to add a person to more than one group, especially to the Favourites and Emergency Contacts groups could be added.
3. The project could be linked to a database containing the contact information so that the user does not have to input all the data every time they use the application.

REFERENCES

1. GeeksForGeeks, Python GUI – tkinter, 2020. [Online].
   Available: https://www.geeksforgeeks.org/python-gui-tkinter/ [Accessed: Dec. 12, 2021].

2. Python Tutorial, Tkinter Treeview. [Online].
   Available: pythontutorial.net/tkinter/tkinter-treeview/ [Accessed: Dec.14, 2021].

3. Tutorialspoint, Python – Tkinter Scrollbar.[Online].
   Available: tutorialspoint.com/python/tk_scrollbar.htm [Accessed: Jan 3, 2022].

4. Python, tkinter – Python interface to Tcl/Tk. [Online].
   Available: https://docs.python.org/3/library/tkinter.html [Accessed: Dec 28, 2021].

5. Python, tkinter – tkinter.ttk – Tk themed widgets. [Online].
   Available: https://docs.python.org/3/library/tkinter.ttk.html [Accessed: Dec 28, 2021]