##### ONLINE ORGAN DONATION APPLICATION

##### A DESIGN PROJECT I REPORT

###### ***Submitted by***

NAVEEN.R (19113102)

SUMANTH KUMAR REDDY.K (19113105)

UMESH CHANDRA.P (19113118)

**Under the guidance of**

**DR KRISHNAVENI R**

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF TECHNOLOGY**

IN

**COMPUTER SCIENCE AND ENGINEERING**



**MAY 2021**



**BONAFIDE CERTIFICATE**

Certified that this Design project I report **“Online Organ Donation”** is the bonafide work of “**NAVEEN.R (19113102), SUMANTH KUMAR REDDY.K (19113105), UMESH CHANDRA.P (19113118)”** who carried out the Design project I work under my supervision during the academic year 2020-21.

**SIGNATURE**

**DR KRISHNAVENI R**

**DESIGN PROJECT**

**COMPUTER SCIENCE AND ENGINEERING**

**INTERNAL EXAMINER EXTERNAL EXAMINER**

Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Designation:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Designation:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Project Viva - voce conducted on \_\_\_\_\_\_\_\_\_

**ACKNOWLEDGMENT**

We thank **Dr.Angelina Geetha, Head of the Department of Computer Science & Engineering** for her strong support and encouragement throughout our course of study.

We express deep gratitude to our Mentor **DR KRISHNAVENI R, DESIGN PROJECT, Department of Computer Science & Engineering** for her constant guidance and continued assistance in completion of this design project.

We thank all the **Faculty members** and **Technical staff** of Department of Computer Science and engineering for their valuable support and suggestions at various stages of design project development.

We are extremely indebted to our family members for their adorable support and care.

Student(s) Name

NAVEEN.R

SUMANTH KUMAR REDDY.K

UMESH CHANDRA.P

**ABSTRACT**

Organ donation is one of the most significant contributions that a person can make towards the society. Organ donation is the process when a person allows an organ of their own to be removed and transplanted to another person legally either by consent while the donor is alive or dead. Donation is the giving of an organ and tissue to help someone that needs a transplant. Transplants can save or transform the life of a person. One organ and tissue donor can help transform the lives of more than 10 people. This relies on donors and their families agreeing to donate their organ and tissue after death. This Application acts as an essential role in saving the lives of human beings and which is also its main aim is to help the user to get the required blood and organs at the correct time. This application provides an easy and fast way to search for organs. The main aim of developing this Application is to provide organs to the people who require organs. The numbers of persons who need organs are increasing in massive amounts day by day. Using this Online Organ Donation Application people can register himself or herself who want to donate organs. To log in the Application, they have to enter their contact information like aadhar card number, mobile number, etc. One deceased organ donor can save up to eight lives. Kidneys, heart, liver, pancreas, intestines, lungs, bones, bone marrow, skin, and corneas are the some common transplantations.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO** |
|  | **ABSTRACT** | 4 |
|  | **LIST OF FIGURES** | 7 |
|  | **LIST OF ABBREVATIONS** | 8 |
|  |  |  |
|  |  |  |
|  |  |  |
| **1.** | **INTRODUCTION** |  |
|  | 1.1 OVERVIEW | 09 |
|  | 1.2 MOTIVATION | 10 |
|  | 1.3 GOAL AND OBJECTIVES | 10 |
|  | 1.4 SCOPE AND MOTIVATION | 11 |
|  | 1.5 ORGANIZATION OF DESIGN PROJECT REPORT | 11 |
|  |  |  |
| **2.** | **LITERATURE REVIEW** |  |
|  | 2.1 REFFERED JOURNAL | 12 |
|  | 2.2 SUMMARY | 13 |
|  |  |  |
| **3.** | **SYSTEM ANALYSIS AND REQUIREMENTS** |  |
|  | 3.1 EXISTING SYSTEM AND DISADVANTAGES | 14 |
|  | 3.2 PROPOSED SYSTEM AND ADVANTAGES | 14 |
|  | 3.3 SYSTEM REQUIREMENTS |  |
|  | 3.3.1 Hardware Requirements | 16 |
|  | 3.3.2 Software Requirements | 16 |
|  |  |  |
| **4.** | **SYSTEM DESIGN** |  |
|  | 4.1 SYSTEM ARCHITECTURE | 17 |
|  | 4.2 MODULAR DESCRIPTION | 19 |
|  | 4.2.1 MODULE 1 | 19 |
|  | 4.2.2 MODULE 2 | 19 |
|  | 4.2.3 MODULE 3 | 20 |
|  | 4.2.4 MODULE 4 | 20 |
|  |  |  |
| **5.** | **IMPLEMENTATION** |  |
|  | 5.1 GENERAL | 21 |
|  | 5.1.1 Main Window | 21 |
|  | 5.1.2 Access Code | 22 |
|  | 5.1.3 Organ Donation Form Window | 22 |
|  |  |  |
| **6.** | **RESULT AND ANALYSIS** | 23 |
|  |  |  |
| **7.** | **CONCLUSION AND FUTURE WORK** |  |
|  | 7.1 CONCLUSION | 24 |
|  | 7.2 FUTURE WORK | 24 |
|  |  |  |
|  | **REFERENCES** | 25 |
|  | **APPENDIX A - CODE SNIPPETS** | 26 |
|  | **APPENDIX B - SAMPLE SCREENSHOTS** | 75 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **TITLE** | **PAGE NO** |
|  |  |  |
| 4.1 | System Architecture | 17 |

6.1 Patients on the waiting List vs 23

Transplants Performed

b.1 Welcome screen and login or signup 75

page

b.2 User login using access codes 75

b.3 Creation of new account 76

b.4 Generating access codes for users by 76

admin

b.5 Organs donation form 77

**LIST OF ABBREVIATIONS**

BOP - Blood and Organ for Patients

GPS - Global Positioning System

OTP - One Time Password

IDLE - Integrated development learning environment

**Chapter 1: Introduction**

* 1. **Overview**

To increase the number of organ donors, especially among underrepresented populations, current approaches include the use of optimized social network interventions, exposing tailored educational content about organ donation to target social media users. Some organs and tissues can be donated by living donors, such as a kidney or part of the liver, part of the pancreas, part of the lungs or part of the intestines, but most donations occur after the donor has died.

This project is based on the getting the organ donation form to fill the donors easily with in less time. First the user should sign-in if the user had an account already by using the access code which is sent by the admin to the user email id else the user will have to sign-up by giving email address, username. After login was success the user will see the donation form. User will have to give the information like contact number, name, aadhar number, blood group etc.

Then the user will have to enter as ‘yes’ if user wants to donate the organ otherwise enter as ‘no’. User will have to enter the same for all the organs based on his/her interest.

After finishing user have to submit the form and user have to click the generate donor card so that user will able to see the donor card.

* 1. **Motivation**

1. To understand the necessity of each organ by all the citizens and to donate organs.

2. To increase the rate of organ donations so that at-least some of the people will get organs transplantation.

3. To provide immediate storage and retrieval of data and information to start the immediate search for the organ donor.

* 1. **Goal and Objectives**

1. Goal

The aim of this project is to develop an application to donate organs without going to any hospital to fill organ donation form. To provide fast and easy accessible application to donors.

1. Objectives

The data is access through the text files and saved in files. It can search the donor details by searching register number.

**1.4 Scope of the Project**

The proposed system mainly focus on provide organ donation form application and to know the necessity of each organs.

1.5 **Organization of Design Project Report**

The overall report revolves around the objective of online organ donation

First chapter deals with introduction of online organ donation. In that we have included overview, motivation, goals and objectives and scope.

Second chapter deals with literature review. In that we include details of every literature survey of we collected,

Third chapter deals with system Analysis and requirements. In this we can see the existing system with disadvantages, proposed system with advantages and system requirements.

Fourth chapter deals with system design. This includes system architecture and modular descriptions.

Fifth module deals with implementation. In this chapter we will see how to implement the system.

Sixth module deals with result and analysis

Seventh chapter deals with conclusion part and future work.

**Chapter 2: Literature Review**

**2.1 REFERRED JOURNAL OR BOOK**

2.1.1 International Journal of Research in Engineering and Technology

This survey was published in the year of 2016 May by Nikita M. Lunawat, Chetan D. Kshirsagar, Ashish A. Gawhande, Rohini M. Rathod, Apurva D. Thool, Shrikant C. Chumble and we saw the : Blood & Organ for Patients (B.O.P), Global Positioning System (GPS) and Android Application.

2.1.2 International Journal of Science, Engineering and Technology Research

This survey was published in the year of 2018 April by S.Arasu, S.Sana Sadaf, K.Shalini and we saw how to identify Organ Donors with Fingerprint based on Image Segmentation Algorithm.

2.1.3 SSRG International Journal of Computer Science and Engineering

This survey was published in the year of 2020 August by K.Pathrakali, V.RupikaThangam, B.SelvaLakshmi, Dr.V.Kavidha and we saw Online Transplant Management System of Blood and Organs.

2.1.4 American Journal of Transplantation

This survey was published in the year of 2016 March by K. Kumar, E. A. King, A. D. Muzaale, J. M. Konel, K. A. Bramstedt, A. B. Massie1, D. L. Segev and A. M. Cameron and we saw brief communication using Smartphone App for Increasing Live Organ Donation.

**2.2 Summary**

This chapter deals with literature survey we have collected the four literatures and we saw the how to use online organ donation and how to design an online organ donation for filling the organ donation form.

**Chapter 3: SYSTEM ANALYSIS AND REQUIREMENTS**

**3.1 Existing System and Disadvantages**

The existing system was donating organs by filling the details of donor and the donor must keep the fingerprint in-order to fill the details for donating organs which should have good internet connectivity for the device(s) to procced further task. Otherwise it may take much longer time to go to next task.

In the hospitals they will provide organ donation form paper for the donors who are willing to donate their organs.

**Disadvantages**

1. Once the user submits the form they can’t edit the form.
2. OTP verification for the user is not available.
3. While user submitting the form if the network is poor then user will have to submit it again.
4. Only one user will be able to login in a single device.
5. One account can’t log-in in multiple devices.
6. Fingerprint type of systems takes much longer time.

**3.2 Proposed System and Advantages**

The proposed method is to create an application so that the organ donors are available easily within the required time. The work done by admin in the interface is

1. Manage donor details, Update donor details.

2. Each register member possesses a user-id and password, which identifies members uniquely.

Each member register as a donor has a user-id and password and if user had already an account then user will have to sign-in using access code which will be sent to user email else user have to sign-up for creating new account. Then user will get a login form if they enter the login details user-id & password. The donor can change password. Donor can fill details in the organ donation form. After filling the details the donor can submit the form and can download the donor card.

**Advantages**

1. We can find the percentile rate of organ donors per year easily.
2. Details of organ donor will be secured.
3. Time consumption is less.
4. Giving quick and accurate information to donate organs by providing organ donation form.
5. Duplication is avoided by using a unique id aadhar card number.

**3.3** **System Requirements**

**3.3.1 Hardware Requirements**

Ram: 4GB

Hard disk: 50Gb (free space required)

Processor: 2.4GHz speed processor

An enhanced keyboard

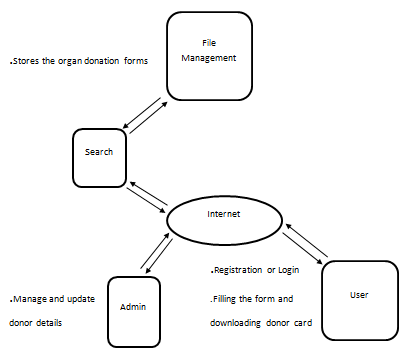
**3.3.1 Software specifications**

Operating System: Windows 7 or >7 version, Mac, Linux

Python idle (from 3.5 version) or Spyder

**Chapter 4: System Design**

**4.1 System Architecture**

****

**Figure 4.1 System Architecture**

The above diagram is the system design for this project Online Organ Donation. In this application if the user had already an account then the user will have to log-in by entering the access-code which is generating by the admins only. Other-wise the user have to sign-up first by the entering the details like Username, Email Address.

After the account was created successfully user will have to fill the details in organ donation form like Phone Number, Gender, Blood Group, Date of Birth etc. and organ’s. Then user can submit the code after filling the details and by clicking the generate donor card option the user will see the donor card.

Only admin type of user’s can manage the edit option for the organ donation form and can update the donor details. And can search the donation forms by using register numbers. Also duplication can be avoided using the donor’s aadhar card number.

By using the file management the organ donation forms will be stored in the files section and all this process can be done only through internet connectivity.

**4.2 MODULAR DESCRIPTION**

List of Modules:

There are four lists of modules in this project online organ donation.

They are:

1. Organ donor or user module
2. Search module
3. User management or Admin module
4. File management module

4.2.1 Organ donor module:

Organ donor module is used to donate the form. i. e, The user should fill the details like personal information which includes User full name, Aadhar Card Number, Email Address, Age, Date of Birth, Gender and Blood Group in form before he/she donates the organs. This is used to give the organs to the particular person who is in need. The user who is ready to donate the organs must register in the organ donation module and he/ she should register and fill the donation form. If the donor wants to donate any organ then user will have to type as ‘yes’ in the given empty box otherwise type as ’no’.

4.2.2 Search module:

Search option is used to search the organs according to the persons. The person who is donating or giving the organs can search regarding to the names of the organs which is given in the search. The hospital admin can search the organs in this module and can know the how many different types of organs are donated. Admin type of users can search using the register number which is different for all users.

4.2.3 User Management Module:

The forms which are submitted by the users will be stored using file management. Admin can update and manage the details of organ donors. This module will be enabled only to the admin type of users. Admin type of users can also present the monthly graph of organ donation form and the percentile rate of donations.

4.2.4 File management module:

All the organ donation forms which are submitted by the users will be stored in the files in the laptop or desktop. All the donation forms can be viewed in this module by admin type of users. By using this stored data admins can be calculated easily percentile rate of donation forms per monthly or yearly.

**Chapter 5: Implementation:**

**5.1 General**

This chapter is the part that puts a planned system into action and examine in details the analysis and design of the Skillmid supermarket system. The present chapter discusses the implementation of the system, highlighting the testing exercise and describing some of the main components of the system's Graphical User Interface. It will give an output from programming language and other tools used to develop our system.

In order to implement the code we need to install some tools for compile and to run the program. We can install either python idle or spyder or jupyter. If we install python idle then we should install numpy using the command ‘pip install numpy’ and for pandas the command is ‘pip install pandas’ in order to avoid some errors or warnings. After typing the command to install numpy if it raises an error that pip is not recognized then we have to add the path in environmental variables where the python idle is installed. Otherwise if we want to install spyder or jupyter then firstly we need to install anaconda navigator. In this navigator we need to launch the spyder or jupyter and there is no need to install pip, numpy and pandas since they are installed by default.

**Main window:**

1. When we run the code by clicking the run module in python idle or in spyder run welcome screen will be displayed by two options sign in and sign up status with user code.
2. If the user is new to this application the click sign-up button in-order to create a new account. Otherwise click sign-in to login by typing the access code which is already sent to user email when the user created the account.

**Access code:**

Admin will generate the access codes for user for log-in which are sent to user email address and it will be kept as privacy. So other user will not know.

**Organ donation form window:**

1. Donor will enter the details like name, phone number, aadhar number etc.

2. Then organs which are donated by the donor will be filled by entering yes or no in the given empty space.

3. After submitting the form by donor, donor will be generated.

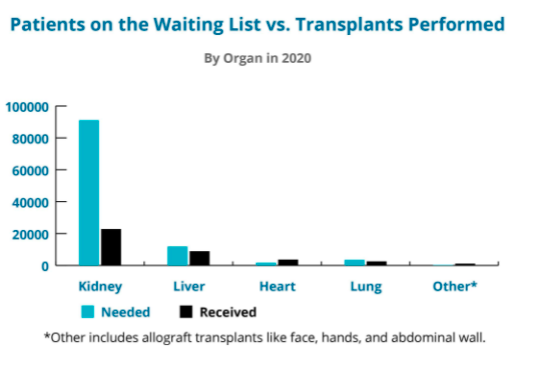
**CHAPTER 6: Result and Analysis**

To reduce the shortcomings of the existing systems for donating organs there is a need to develop a new system that could upgrade the status of the current system which is manual and slow process. The new system should be concern with offering the requirements of the customer and the workers, the system should be reliable, easier, fast, and more informative.

We know that there are so many technologies now-a-days like GPS system, Finger print but this system will reduce the time consumption which is user friendly and don’t need more technical knowledge. You can easily operate the operations in this system.

Organ Donation is donating a donor's organs like heart, liver, kidneys, intestines, lungs, and pancreas, after the donor dies, for the purpose of transplanting them into another person who is in need of an organ. The organ of the donor can be transplanted to the patient who needs it urgently.

107,000+ number of men, women, and children on the national transplant waiting list as of February 2021. 39,000 transplants were performed in 2020. 17 people die each day waiting for an organ transplant.



**Fig 6.1 Patients on the waiting List vs Transplants Performed**

**CHAPTER 7: Conclusion and Future Work**

**7.1 Conclusion**

In conclusion, this Online Organ Donation application provides a better, faster and effective way for the citizens to communicate with the organ donor for organ requirement in times of medical emergency. All the information is stored using file management. Duplication is avoided by using a unique id aadhar card number and fingerprint.

**7.2 Future Work:**

In the future we will bring a significant change in our locality. A new, updated and expanded edition of our project is to implement the cloud computing function to increase the rise of organ donors, and to translate the system into regional language. This work may be extended to interconnect the entire organ donor’s societies in the India. Collecting customer reviews directly from the application.

**References**

[1] K. Kumar1, E. A. King1, A. D. Muzaale1, J. M. Konel, K. A. Bramstedt, A. B. Massie1, D. L. Segev and A. M. Cameron, “A Smartphone App for Increasing Live Organ Donation”, American Journal of Transplantation 2016; 16: 3548–3553 Wiley Periodicals Inc., March 2016.

[2] Kathy Knox, Joy Parkinson, Bo Pang, Haruka Fujihira, Patricia David and Sharyn Rundle-Thiele, “A Systematic Literature Review and Research Agenda for Organ Donation Decision Communication”, Progress in Transplantation 2017, Vol. 27(3) 309-320.

[3] Francisco Javier Mercado-Martínez, César Padilla-Altamira, Blanca Díaz-Medina, Carlos SánchezPimienta, “Views of health care personnel on organ donation and transplantation: A literature review”, Text Context Nursing, Florianópolis, 2015 Abr-Jun; 24(2): 574-83.

[4] Vijayalakshmi Poreddi, BV Katyayani, Sailaxmi Gandhi, Rohini Thimmaiah, Suresh Badamath, ”Attitudes, knowledge, and willingness to donate organs among Indian nursing students”, Saudi Journal of Kidney Diseases and Transplantation, Vol.27, Issue6, November 2016.

[5] Raktim Pratim Tamuli, Smritimala Sarmah, Bishwajeet Saikia, “Organ donation – Attitude and undergraduates and postgraduates of North-East India”, Journal of Family Medicine and Primary Care, Vol.8, Issue1, January 2019.

**Source Code**

import os

import random

from tkinter import \*

from tkinter import Toplevel, Button, Tk, Menu

from tkinter import messagebox

# from PIL import ImageTk,Image

#welcome window

root = Tk()

#main window and functions

def main():

root = Tk()

def bill\_no():

try:

z=open("reg\_no","x")

z.close()

z = open("reg\_no", "w")

z.write("1")

z.close()

except:

z=open("reg\_no","r")

e=int(z.readline())

e3.delete(0,END)

z.close()

e3.insert(0,"{}".format(e))

e+=1

z=open("reg\_no","w")

z.write("{}".format(e))

z.close()

def clear():

e1.delete(0,END)

e2.delete(0,END)

e3.delete(0,END)

e4.delete(0,END)

e5.delete(0,END)

e6.delete(0,END)

e7.delete(0,END)

e8.delete(0,END)

e9.delete(0,END)

e10.delete(0,END)

e11.delete(0,END)

e12.delete(0,END)

e13.delete(0,END)

e14.delete(0,END)

e15.delete(0,END)

e16.delete(0,END)

e17.delete(0,END)

e18.delete(0,END)

e19.delete(0,END)

e20.delete(0,END)

e21.delete(0,END)

e22.delete(0,END)

e23.delete(0,END)

e24.delete(0,END)

e25.delete(0,END)

e26.delete(0,END)

e27.delete(0,END)

t1.delete(1.0,END)

bill\_no()

#Generate form in organ donation form area and store it in a different file with a unique reg no.

def gen\_bill():

t1.insert(END, " \n "

" Online Organ Donation\n\n\n\tI AM AN ORGAN DONOR\n\nName : Shankar\nContact No:9441337299")

b = os.getcwd()

os.chdir("{}/bill".format(b))

z = open("{}".format(e3.get()),"r")

for i in range(20):

k = z.readline()

t1.insert(END, "{}".format(k))

t1.insert(END, "\n \n Total : {0}".format(float(e4.get()[3:])+float(e6.get()[3:])+float(e8.get()[3:])))

z.close()

z = open("{}".format(e3.get()), "a")

z.write("Total : {0}".format(float(e4.get()[3:])+float(e6.get()[3:])+float(e8.get()[3:])))

z.close()

os.chdir("{}".format(b))

#search form using a unique reg. no. and displays.

def sea():

t1.delete(1.0, END)

try:

b=os.getcwd()

os.chdir("{}/bill".format(b))

z=open("{}".format(e3.get()),"r")

for i in range(20):

k=z.readline()

t1.insert(END,"{}".format(k))

os.chdir("{}".format(b))

except:

messagebox.showinfo("error","Specified Donor is Not available.")

#generate donor card

def total():

global d, e

e = d = f = w = 0

e1.get()

e2.get()

e3.get()

co1 = [e10.get(), e11.get(), e12.get(), e13.get(), e14.get(), e15.get()]

g1 = [e16.get(), e17.get(), e18.get(), e19.get(), e20.get(), e21.get()]

col1 = [e22.get(), e23.get(), e24.get(), e25.get(), e26.get(), e27.get()]

col2 = [e22, e23, e24, e25, e26, e27]

g2 = [e16, e17, e18, e19, e20, e21]

co2 = [e10, e11, e12, e13, e14, e15]

g = ["rc", "fo", "da", "wh", "su", "te"]

c = ["ba", "fc", "fw", "hs", "hg", "bl"]

co = ["mz", "coc", "fru", "thum", "lim", "spr"]

r = ["grot", "cost", "colt"]

l = 0

for i in g:

z = open(i, "r")

e = int(z.readline())

z.close()

n = e \* int(g1[l])

l += 1

d += n

z = open(r[0], "r")

u = float(z.readline())

e7.insert(1, "{}".format(u))

z.close()

p = d \* (u / 100)

s = p + d

e6.insert(1, "Rs.{}".format(s))

l = 0

for i in g:

z = open(i, "r")

e = int(z.readline())

z.close()

n = e \* int(co1[l])

l += 1

w += n

z = open(r[1], "r")

u = float(z.readline())

e5.insert(1, "{}".format(u))

z.close()

s = w \* (u / 100)

q = s + w

e4.insert(1, "Rs.{}".format(q))

l = 0

for i in co:

z = open(i, "r")

e = int(z.readline())

z.close()

n = e \* int(col1[l])

l += 1

f += n

z = open(r[2], "r")

u = float(z.readline())

e9.insert(1, "{}".format(u))

z.close()

o = f \* (u / 100)

e = o + f

e8.insert(1, "Rs.{}".format(e))

z = open("bill\_no", "r")

s = z.readline()

z.close()

try:

b = os.getcwd()

os.mkdir("{}/bill".format(b))

os.chdir("{}/bill".format(b))

z = open("{}".format(e3.get()), "w")

n = [e10.get(), e11.get(), e12.get(), e13.get(), e14.get(), e15.get(),

e16.get(), e17.get(), e18.get(), e19.get(), e20.get(), e21.get(), e22.get(), e23.get(), e24.get(),

e25.get(), e26.get(), e27.get()]

m = ["Aadhar Card No", "Email", "Age", "Gender", "Blood Group"]

z.write("Name :{0} \nContact :{1}\nReg no.{2}\n\n".format(e1.get(), e2.get(), s))

z.write("\n")

l = 0

for i in m:

z.write("{0}\t\t{1}\n".format(i, n[l]))

l += 1

z.close()

os.chdir("{}".format(b))

except:

os.chdir("{}/bill".format(b))

z = open("{}".format(e3.get()), "w")

n = [e10.get(), e11.get(), e12.get(), e13.get(), e14.get(), e15.get(),

e16.get(), e17.get(), e18.get(), e19.get(), e20.get(), e21.get(), e22.get(), e23.get(), e24.get(),

e25.get(), e26.get(), e27.get()]

m = ["Aadhar Card No", "Email", "Age", "Gender", "Blood Group"]

z.write("Name :{0} \nContact :{1}\nReg no.{2}\n\n".format(e1.get(), e2.get(), s))

z.write("\n")

l = 0

for i in m:

z.write("{0}\t\t{1}\n".format(i, n[l]))

l += 1

z.close()

os.chdir("{}".format(b))

l = 0

for i in co:

col2[l].delete(0, END)

z = open(i, "r")

e = z.readline()

col2[l].insert(1, "Rs.{}".format(e))

z.close()

l += 1

l = 0

for i in g:

g2[l].delete(0, END)

z = open(i, "r")

e = z.readline()

g2[l].insert(1, "Rs.{}".format(e))

z.close()

l += 1

l = 0

for i in c:

co2[l].delete(0, END)

z = open(i, "r")

e = z.readline()

co2[l].insert(1, "Rs.{}".format(e))

z.close()

l += 1

root.geometry("1280x720")

root.minsize(960, 540)

root.configure(bg="green")

root.title("ORGAN DONATION FORM")

# top frame

topframe = LabelFrame(root, text="Details", bg="grey", font=("comicsansms", 12, "bold"), fg="yellow",

height="60", relief=SUNKEN, borderwidth=5)

l2 = Label(topframe, text="Donor Name", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(x=10, y=5)

e1 = Entry(topframe, relief=SUNKEN, borderwidth=5)

e1.place(x=125, y=-1)

l3 = Label(topframe, text="Contact No.", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(x=429, y=5)

e2 = Entry(topframe, relief=SUNKEN, borderwidth=5)

e2.place(x=515, y=-1)

l4 = Label(topframe, text="Reg No.", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(

x=860, y=5)

e3 = Entry(topframe, relief=SUNKEN, borderwidth=5)

e3.place(x=923, y=-1)

try:

z = open("bill\_no", "x")

z.close()

z = open("bill\_no", "w")

z.write("1")

z.close()

except:

z = open("bill\_no", "r")

e = int(z.readline())

z.close()

e3.insert(0, "{}".format(e))

e += 1

z = open("bill\_no", "w")

z.write("{}".format(e))

z.close()

btn1 = Button(topframe, activebackground="black", activeforeground="white", text="Search",

font=("comicsansms", 12, "bold"), relief=SUNKEN, borderwidth=3,command=sea).place(x=1115, y=-3)

topframe.pack(side=TOP, fill=X, padx=10, pady=10)

bottomframe = LabelFrame(root, text="Generating Donor Card", bg="grey", fg="yellow", font=("comicsansms", 12, "bold"),

height="150", relief=SUNKEN, borderwidth=5)

btn2 = Button(bottomframe, activebackground="black", activeforeground="white", text="Submit",

font=("comicsansms", 12, "bold"), relief=SUNKEN, height=2, width=6, borderwidth=9,command=total).place(x=820,

y=30)

btn3 = Button(bottomframe, activebackground="black", activeforeground="white", text="Generate\nDonor Card",

font=("comicsansms", 12, "bold"), relief=SUNKEN, height=2, width=12, borderwidth=9,command=gen\_bill).place(x=940,

y=30)

btn4 = Button(bottomframe, activebackground="black", activeforeground="white", text="Clear",

font=("comicsansms", 12, "bold"), relief=SUNKEN, height=2, width=6, borderwidth=9,command=clear).place(x=1110,

y=30)

btn5 = Button(bottomframe, activebackground="black", activeforeground="white", text="Exit",

font=("comicsansms", 12, "bold"), relief=SUNKEN, height=2, width=5, borderwidth=9,

command=root.destroy).place(x=1220,

y=30)

bottomframe.pack(side=BOTTOM, fill=X, padx=10, pady=10)

cntrframe1 = LabelFrame(root, text="Donor Details", bg="grey", fg="yellow", font=("comicsansms", 12, "bold"),

height="70", width="325", relief=SUNKEN, borderwidth=5)

l13 = Label(cntrframe1, text="Aadhar No", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(x=10, y=15)

e10 = Entry(cntrframe1, relief=SUNKEN, borderwidth=5)

e10.place(x=125, y=11)

l14 = Label(cntrframe1, text="Email Address", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(x=10, y=80)

e11 = Entry(cntrframe1, relief=SUNKEN, borderwidth=5)

e11.place(x=125, y=80)

l15 = Label(cntrframe1, text="Date of Birth", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(x=10, y=145)

e12 = Entry(cntrframe1, relief=SUNKEN, borderwidth=5)

e12.place(x=125, y=145)

l16 = Label(cntrframe1, text="Age", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(x=10, y=210)

e13 = Entry(cntrframe1, relief=SUNKEN, borderwidth=5)

e13.place(x=125, y=210)

l17 = Label(cntrframe1, text="Gender", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(x=10, y=275)

e14 = Entry(cntrframe1, relief=SUNKEN, borderwidth=5)

e14.place(x=125, y=275)

l18 = Label(cntrframe1, text="Blood Group", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(x=10, y=340)

e15 = Entry(cntrframe1, relief=SUNKEN, borderwidth=5)

e15.place(x=125, y=340)

cntrframe1.pack(side=LEFT, fill=Y, padx=10, pady=10)

cntrframe2 = LabelFrame(root, text="Organs", bg="grey", fg="yellow", font=("comicsansms", 12, "bold"), height="70",

width="325", relief=SUNKEN, borderwidth=5)

l20 = Label(cntrframe2, text="Eyes", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(

x=10, y=15)

e16 = Entry(cntrframe2, relief=SUNKEN, borderwidth=5)

e16.place(x=125, y=11)

l21 = Label(cntrframe2, text="Kidney", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(x=10, y=80)

e17 = Entry(cntrframe2, relief=SUNKEN, borderwidth=5)

e17.place(x=125, y=80)

l22 = Label(cntrframe2, text="Heart", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(

x=10, y=145)

e18 = Entry(cntrframe2, relief=SUNKEN, borderwidth=5)

e18.place(x=125, y=145)

l23 = Label(cntrframe2, text="Liver", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(

x=10, y=210)

e19 = Entry(cntrframe2, relief=SUNKEN, borderwidth=5)

e19.place(x=125, y=210)

l24 = Label(cntrframe2, text="Lungs", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(

x=10, y=275)

e20 = Entry(cntrframe2, relief=SUNKEN, borderwidth=5)

e20.place(x=125, y=275)

l25 = Label(cntrframe2, text="Intestines", bg="grey", fg="black", font=("comicsansms", 12, "bold"), relief=FLAT).place(

x=10, y=340)

e21 = Entry(cntrframe2, relief=SUNKEN, borderwidth=5)

e21.place(x=125, y=340)

e22 = Entry(cntrframe2, relief=SUNKEN, borderwidth=5)

e22.place(x=125, y=210)

l26 = Label(cntrframe2, text="Pancreas", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(

x=10, y=275)

cntrframe2.pack(side=LEFT, fill=Y, padx=10, pady=10)

cntrframe3 = LabelFrame(root, text="Donor Card", height="70", width="325", bg="white", fg="BLACK",

font=("comicsansms", 20, "bold"), relief=SUNKEN, borderwidth=5)

sb1 = Scrollbar(cntrframe3)

sb1.pack(side=RIGHT, fill=Y)

t1 = Text(cntrframe3, height=700, width=35, yscrollcommand=sb1.set)

t1.pack(fill=Y, padx=6, pady=6)

sb1.config(command=t1.yview)

cntrframe3.pack(side=RIGHT, fill=Y, padx=10, pady=10)

root.mainloop()

#Signup window

def sgnup():

global oge, cpe, cpe2, nme

root2 = Tk()

#window of changing the details in the form

def item\_values():

root5 = Tk()

def changed():

a = e16.get()

b = e17.get()

c = e18.get()

d = e19.get()

e = e20.get()

f = e21.get()

g = e22.get()

h = e23.get()

i = e24.get()

j = e25.get()

k = e26.get()

l = e27.get()

m = e28.get()

n = e29.get()

o = e30.get()

p = e31.get()

q = e32.get()

r = e33.get()

s = e34.get()

t = e35.get()

u = e36.get()

z = open("rc", "w")

z.write("{}".format(a))

z.close()

v = open("fo", "w")

v.write("{}".format(b))

v.close()

w = open("da", "w")

w.write("{}".format(c))

w.close()

x = open("wh", "w")

x.write("{}".format(d))

x.close()

y = open("su", "w")

y.write("{}".format(e))

y.close()

aa = open("te", "w")

aa.write("{}".format(f))

aa.close()

bb = open("grot", "w")

bb.write("{}".format(g))

bb.close()

cc = open("cost", "w")

cc.write("{}".format(h))

cc.close()

dd = open("ba", "w")

dd.write("{}".format(i))

dd.close()

ee = open("fc", "w")

ee.write("{}".format(j))

ee.close()

ff = open("fw", "w")

ff.write("{}".format(k))

ff.close()

gg = open("hs", "w")

gg.write("{}".format(l))

gg.close()

hh = open("hg", "w")

hh.write("{}".format(m))

hh.close()

ii = open("bl", "w")

ii.write("{}".format(n))

ii.close()

jj = open("colt", "w")

jj.write("{}".format(o))

jj.close()

kk = open("mz", "w")

kk.write("{}".format(p))

kk.close()

tt = open("coc", "w")

tt.write("{}".format(q))

tt.close()

ll = open("fru", "w")

ll.write("{}".format(r))

ll.close()

pp = open("thum", "w")

pp.write("{}".format(s))

pp.close()

ww = open("lim", "w")

ww.write("{}".format(t))

ww.close()

qq2 = open("spr", "w")

qq2.write("{}".format(u))

qq2.close()

root5.geometry("1280x720")

root5.minsize(965, 460)

root5.title("Change form Details ")

root5.configure(bg="black")

f12 = LabelFrame(root5, text="Donor Details", height="251", width="325", bg="grey", fg="yellow",

font=("comicsansms", 12, "bold"), relief=SUNKEN, borderwidth=5)

l43 = Label(f12, text="Taxs.", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(x=10, y=80)

e22 = Entry(f12, relief=SUNKEN, borderwidth=5)

e22.place(x=125, y=80)

f12.place(x=10, y=440)

f9 = LabelFrame(root5, text="Aadhar No", height="420", width="325", bg="grey", fg="yellow",

font=("comicsansms", 12, "bold"), relief=SUNKEN, borderwidth=5)

l20 = Label(f9, text="Email", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(

x=10, y=15)

e16 = Entry(f9, relief=SUNKEN, borderwidth=5)

e16.place(x=125, y=11)

l21 = Label(f9, text="DOB", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(x=10, y=80)

e17 = Entry(f9, relief=SUNKEN, borderwidth=5)

e17.place(x=125, y=80)

l22 = Label(f9, text="Age", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(

x=10, y=145)

e18 = Entry(f9, relief=SUNKEN, borderwidth=5)

e18.place(x=125, y=145)

l23 = Label(f9, text="Gender", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(

x=10, y=210)

e19 = Entry(f9, relief=SUNKEN, borderwidth=5)

e19.place(x=125, y=210)

l24 = Label(f9, text="Blood Group", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(

x=10, y=275)

e21 = Entry(f9, relief=SUNKEN, borderwidth=5)

e21.place(x=125, y=340)

f9.place(x=10, y=10)

f13 = LabelFrame(root5, text="Organs form", height="251", width="325", bg="grey", fg="yellow",

font=("comicsansms", 12, "bold"), relief=SUNKEN, borderwidth=5)

l44 = Label(f13, text="", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(x=10, y=80)

e23 = Entry(f13, relief=SUNKEN, borderwidth=5)

e23.place(x=125, y=80)

f13.place(x=356, y=440)

f10 = LabelFrame(root5, text="Organs", height="420", width="325", bg="grey", fg="yellow",

font=("comicsansms", 12, "bold"), relief=SUNKEN, borderwidth=5)

l13 = Label(f10, text="Eye", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(x=10, y=15)

e24 = Entry(f10, relief=SUNKEN, borderwidth=5)

e24.place(x=125, y=11)

l14 = Label(f10, text="Kidney", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(x=10, y=80)

e25 = Entry(f10, relief=SUNKEN, borderwidth=5)

e25.place(x=125, y=80)

l15 = Label(f10, text="Heart", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(x=10, y=145)

e26 = Entry(f10, relief=SUNKEN, borderwidth=5)

e26.place(x=125, y=145)

l16 = Label(f10, text="Liver", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(x=10, y=210)

e27 = Entry(f10, relief=SUNKEN, borderwidth=5)

e27.place(x=125, y=210)

l17 = Label(f10, text="Lungs", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(x=10, y=275)

e28 = Entry(f10, relief=SUNKEN, borderwidth=5)

e28.place(x=125, y=275)

l18 = Label(f10, text="Intestines", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(x=10, y=340)

e29 = Entry(f10, relief=SUNKEN, borderwidth=5)

e29.place(x=125, y=340)

f10.place(x=356, y=10)

btn10 = Button(root5, activebackground="black", activeforeground="white", text="Change",

font=("comicsansms", 12, "bold"), relief=SUNKEN, borderwidth=3, command=changed).place(x=1140, y=600)

btn9 = Button(root5, activebackground="black", activeforeground="white", text="Back",

font=("comicsansms", 12, "bold"), relief=SUNKEN, borderwidth=3, command=root5.destroy).place(x=1220, y=600)

root5.mainloop()

#About window

def about():

root7 = Tk()

root7.geometry("1280x720")

root7.minsize(965, 460)

root7.title("ABOUT")

root7.configure(bg="black")

photo = PhotoImage(file="2.png")

x\_lable = Label(image=photo)

Label.pack(self=x\_lable)

l47 = Label(root7, text="OOD : ", bg="black", fg="Grey", font=("comicsansms", 30, "bold"),

relief=FLAT).place(x=650, y=10)

l46 = Label(root7, text="Online Organ Donation\n Application", bg="black", fg="Grey", font=("comicsansms", 30, "bold"),

relief=FLAT).place(x=560, y=60)

l45 = Label(root7, text="\n\nSafe Surffing Over Internet, Be Like Anonymous", bg="black", fg="Grey",

font=("comicsansms", 15, "bold"),

relief=FLAT).place(x=490, y=110)

l44 = Label(root7, text="-Organ donation is the process when a person allows an organ of their own to be removed\nand transplanted to another person legally either",bg="black", fg="Grey",

font=("comicsansms", 15, "bold"),

relief=FLAT).place(x=210, y=160)

l43 = Label(root7, text="by consent while the donor is alive or dead.\n-This app provides quick and accurate information. ",

bg="black", fg="Grey", font=("comicsansms", 15, "bold"),

relief=FLAT).place(x=230, y=210)

l48 = Label(root7, text="-This can make a query from people into action, thus saving multiple lives.\n-Everyone has the potential to be an organ, eye and tissue donor.", bg="black", fg="Grey",

font=("comicsansms", 15, "bold"),

relief=FLAT).place(x=250, y=260)

l49 = Label(root7, text="-One person can save and heal up to 60 lives through organ, eye and tissue donation!.", bg="black", fg="Grey",

font=("comicsansms", 15, "bold"),

relief=FLAT).place(x=230, y=310)

l50 = Label(root7, text="-We can find the percentile rate of organ donors per year easily.", bg="black", fg="Grey",

font=("comicsansms", 15, "bold"),

relief=FLAT).place(x=320, y=360)

l50 = Label(root7, text=".", bg="black", fg="Grey",

font=("comicsansms", 15, "bold"),

relief=FLAT).place(x=450, y=410)

btn45 = Button(root7, text="Back", bg="black", fg="grey", command=root7.destroy).place(x=1100, y=620)

root7.mainloop()

#Admin window

def Admin():

if oge.get() != "" and nme.get() != "" and cpe.get() != "":

if cpe.get() == cpe2.get():

root = Tk()

root2.destroy()

root.geometry("1280x720")

root.minsize(965, 460)

root.title("USER CONSOLE")

root.configure(bg="black")

#Add Access code window

def Add\_access\_code():

root5 = Tk()

root5.maxsize(400, 400)

root5.minsize(400, 400)

root5.title("ACCESS CODE")

root5.configure(bg="black")

global a, b, c, d, x

a, b, c, d, x = [], [], [], [], ""

def Add(x):

list1.insert(END, "{}: {}".format(nme.get(), x))

try:

f = open("assigned\_name", "x")

t = open("access\_code", "x")

if f:

if t:

f.close()

f = open("assigned\_name", "a")

f.write("{}\n".format(nme.get()))

f.close()

t = open("access\_code", "a")

t.write("{}\n".format(x))

t.close()

listbox2.insert(END, x)

listbox.insert(END, nme.get())

except:

print("already created")

f = open("assigned\_name", "a")

f.write("{}\n".format(nme.get()))

f.close()

t = open("access\_code", "a")

t.write("{}\n".format(x))

t.close()

listbox2.insert(END, x)

listbox.insert(END, nme.get())

#for generating the random Access code for user login

def generate():

global a, b, c, d, x

a, b, c, d, x = [], [], [], [], ""

o = 65

for i in range(o, o + 26):

p = o + 33

for j in range(p - 1, p):

if o < 75:

for g in range(o - 64):

c.append(g)

l = chr(j)

d.append(l)

k = chr(i)

a.append(k)

o += 1

if o == 92:

break

for i in range(2):

n = random.choice(a)

h = random.choice(c)

f = random.choice(d)

b.append(h)

b.append(f)

b.append(n)

for i in range(len(b)):

x = str(x + "{}".format(b[i]))

e1.insert(1, x)

Add(x=x)

f5 = Frame(root5, width="200", bg="grey", borderwidth=2)

l99 = Label(f5, text="Name:").place(x=10, y=1)

nme = Entry(f5, borderwidth=2)

nme.place(x=20, y=30)

e1 = Entry(f5, borderwidth=2)

e1.place(x=20, y=79)

btn9 = Button(f5, text="Generate", command=generate).place(x=20, y=109)

f5.pack(side=RIGHT, fill=Y)

f6 = Frame(root5, width="200", relief=SUNKEN, bg="grey", borderwidth=2)

list1 = Listbox(f6, height="90", width="200")

list1.pack(fill=Y, padx=6, pady=6)

f6.pack(side=LEFT, fill=Y)

root5.mainloop()

def dec():

root.destroy()

main()

menubar = Menu(root, bg="grey", font=("comicsansms", 12, "bold"))

Set = Menu(menubar, tearoff=0)

Set.add\_command(label="Access Code", font=("comicsansms", 12, "bold"), command=Add\_access\_code)

Set.add\_separator()

Set.add\_command(label="Exit", font=("comicsansms", 12, "bold"), command=root.destroy)

menubar.add\_cascade(label="Set", menu=Set)

Bill = Menu(menubar, tearoff=0)

Bill.add\_command(label="Donation Form", font=("comicsansms", 12, "bold"), command=dec)

menubar.add\_cascade(label="Organ's Donation Form", menu=Bill)

Bill.add\_separator()

help = Menu(menubar, tearoff=0)

help.add\_command(label="About", font=("comicsansms", 12, "bold"), command=about)

menubar.add\_cascade(label="Help", menu=help)

root.config(menu=menubar)

f10 = Frame(root, height="10", bg="grey", relief=SUNKEN, borderwidth=5)

l33 = Label(f10, text="List of Access Code", bg="grey", fg="black",

font=("comicsansms", 12, "bold"),

relief=FLAT).pack()

f10.pack(side=TOP, fill=X, padx=10, pady=10)

f12 = Frame(root, height="30", bg="grey")

l34 = Label(f12, text="Assigned To", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(x=50, y=10)

l35 = Label(f12, text="Access Code", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(x=900, y=10)

f12.pack(side=TOP, fill=X, padx=10)

f11 = Frame(root, height="800", bg="grey", borderwidth=5)

listbox = Listbox(f11, height=900, width=100)

listbox.pack(side=LEFT, fill=Y, padx=6, pady=0)

listbox2 = Listbox(f11, height=900, width=100)

try:

f = open("assigned\_name", "r")

z = f.readlines()

for i in z:

k = i

listbox.insert(1, k)

f.close()

t = open("access\_code", "r")

j = t.readlines()

for i in j:

k = i

listbox2.insert(1, k)

except:

messagebox.showinfo("information", "no Access code")

listbox2.pack(side=RIGHT, fill=Y, padx=6, pady=0)

f11.pack(side=TOP, fill=X, padx=10)

btn = Button(f12, text="Delete Name",relief=FLAT, command=lambda listbox=listbox: listbox.delete(ANCHOR))

btn.place(x=450, y=1)

btn90 = Button(f12, text="Delete Access code",relief=FLAT,

command=lambda listbox2=listbox2: listbox2.delete(ANCHOR))

btn90.place(x=1150, y=1)

root.mainloop()

else:

messagebox.showinfo("Warning", "Conform Password ")

else:

messagebox.showinfo("warning", "Fill the details:")

#for creation of admin account

def create\_ac():

try:

f = open("admn", "x").close()

t = open("pa", "x").close()

y = open("or", "x").close()

og = oge.get()

nam = nme.get()

pas = cpe.get()

pa2 = cpe2.get()

t = open("or", "r")

g = t.readline()

f = open("admn", "r")

d = f.readline()

if nam != d and og != g:

f.close()

t.close()

if pas == pa2:

y = open("or", "w")

f = open("admn", "w")

t = open("pa", "w")

f.write(nam)

t.write(pas)

y.write(og)

f.close()

t.close()

y.close()

messagebox.showinfo("Information", "Created User Account")

Admin()

else:

messagebox.showinfo("Error", "Confirm Your Password")

else:

f.close()

t.close()

messagebox.showinfo("Error", "Account Already Exist")

except:

og = oge.get()

nam = nme.get()

pas = cpe.get()

pa2 = cpe2.get()

t = open("or", "r")

g = t.readline()

f = open("admn", "r")

d = f.readline()

if nam != d and og != g:

f.close()

t.close()

if pas == pa2:

y = open("or", "w")

f = open("admn", "w")

t = open("pa", "w")

f.write(nam)

t.write(pas)

y.write(og)

f.close()

t.close()

y.close()

messagebox.showinfo("Information", "Created User Account")

Admin()

else:

messagebox.showinfo("Error", "Confirm Your Password")

else:

f.close()

t.close()

messagebox.showinfo("Error", "Account Already Exist")

root2.title("")

# root1.minsize(965, 460)

# root1.maxsize(965, 460)

f2 = Frame(root2, bg="black", height=460, width=965).pack()

# root1.configure(bg="black")

l38 = Label(f2, text="Creating User Account.", bg="black", fg="grey",

font=("comicsansms", 30, "bold")).place(x=250, y=80)

l39 = Label(f2, text="Email Address:", bg="black", fg="grey", font=("comicsansms", 20, "bold"),

relief=FLAT).place(

x=250, y=170)

oge = Entry(f2, relief=SUNKEN, borderwidth=5)

oge.place(x=520, y=170)

l42 = Label(f2, text="Name:", bg="black", fg="grey", font=("comicsansms", 20, "bold"),

relief=FLAT).place(

x=250, y=220)

nme = Entry(f2, relief=SUNKEN, borderwidth=5)

nme.place(x=520, y=220)

l40 = Label(f2, text="Create Password:", bg="black", fg="grey", font=("comicsansms", 20, "bold"),

relief=FLAT).place(

x=250, y=270)

cpe = Entry(f2, show="\*", relief=SUNKEN, borderwidth=5)

cpe.place(x=520, y=270)

l41 = Label(f2, text="Confirm Password:", bg="black", fg="grey", font=("comicsansms", 20, "bold"),

relief=FLAT).place(x=250, y=320)

cpe2 = Entry(f2, show="\*", relief=SUNKEN, borderwidth=5)

cpe2.place(x=520, y=320)

btn8 = Button(f2, text="Create Account", bg="Grey", fg="Black", font=("comicsansms", 20, "bold"),

command=create\_ac).place(x=620, y=370)

root2.mainloop()

#Signin window

def sgnin():

root1 = Tk()

#destroying root1 and call signup window

def dis():

root1.destroy()

sgnup()

#call Admin login window from signin screen and destroy root1

def dis3():

messagebox.showinfo("Information", "User Login Success.")

root = Tk()

root1.destroy()

root.geometry("1280x720")

root.minsize(965, 460)

root.title("USER CONSOLE")

root.configure(bg="black")

#For making change in form

def item\_values():

root5 = Tk()

# write changes in a file

def changed():

a = e16.get()

b = e17.get()

c = e18.get()

d = e19.get()

e = e20.get()

f = e21.get()

g = e22.get()

h = e23.get()

i = e24.get()

j = e25.get()

k = e26.get()

l = e27.get()

m = e28.get()

n = e29.get()

o = e30.get()

p = e31.get()

q = e32.get()

r = e33.get()

s = e34.get()

t = e35.get()

u = e36.get()

z = open("rc", "w")

z.write("{}".format(a))

z.close()

v = open("fo", "w")

v.write("{}".format(b))

v.close()

w = open("da", "w")

w.write("{}".format(c))

w.close()

x = open("wh", "w")

x.write("{}".format(d))

x.close()

y = open("su", "w")

y.write("{}".format(e))

y.close()

aa = open("te", "w")

aa.write("{}".format(f))

aa.close()

bb = open("grot", "w")

bb.write("{}".format(g))

bb.close()

cc = open("cost", "w")

cc.write("{}".format(h))

cc.close()

dd = open("ba", "w")

dd.write("{}".format(i))

dd.close()

ee = open("fc", "w")

ee.write("{}".format(j))

ee.close()

ff = open("fw", "w")

ff.write("{}".format(k))

ff.close()

gg = open("hs", "w")

gg.write("{}".format(l))

gg.close()

hh = open("hg", "w")

hh.write("{}".format(m))

hh.close()

ii = open("bl", "w")

ii.write("{}".format(n))

ii.close()

jj = open("colt", "w")

jj.write("{}".format(o))

jj.close()

kk = open("mz", "w")

kk.write("{}".format(p))

kk.close()

tt = open("coc", "w")

tt.write("{}".format(q))

tt.close()

ll = open("fru", "w")

ll.write("{}".format(r))

ll.close()

pp = open("thum", "w")

pp.write("{}".format(s))

pp.close()

ww = open("lim", "w")

ww.write("{}".format(t))

ww.close()

qq2=open("spr","w")

qq2.write("{}".format(u))

qq2.close()

root5.geometry("1280x720")

root5.minsize(965, 460)

btn10 = Button(root5, activebackground="black", activeforeground="white", text="Change",

font=("comicsansms", 12, "bold"), relief=SUNKEN, borderwidth=3, command=changed).place(x=1140, y=600)

btn9 = Button(root5, activebackground="black", activeforeground="white", text="Back",

font=("comicsansms", 12, "bold"), relief=SUNKEN, borderwidth=3, command=root5.destroy).place(

x=1220, y=600)

root5.mainloop()

#this is just About page

def about():

root7 = Tk()

root7.geometry("1280x720")

root7.minsize(965, 460)

root7.title("ABOUT")

root7.configure(bg="black")

photo = PhotoImage(file="2.png")

x\_lable = Label(image=photo)

Label.pack(self=x\_lable)

l47 = Label(root7, text="OOD", bg="black", fg="Grey", font=("comicsansms", 30, "bold"),

relief=FLAT).place(x=650, y=10)

l46 = Label(root7, text="Online Organ Donation", bg="black", fg="Grey", font=("comicsansms", 30, "bold"),

relief=FLAT).place(x=560, y=60)

l45 = Label(root7, text="Safe Surffing Over Internet, Be Like Anonymous", bg="black", fg="Grey",

font=("comicsansms", 15, "bold"),

relief=FLAT).place(x=490, y=110)

l44 = Label(root7, text="Organ donation is the process when a person allows an organ of their"

" own to be removed and transplanted to another person legally either", bg="black", fg="Grey",

font=("comicsansms", 15, "bold"),

relief=FLAT).place(x=210, y=160)

l43 = Label(root7, text=" by consent while the donor is alive or dead."

" It provides quick and accurate information. ",

bg="black", fg="Grey", font=("comicsansms", 15, "bold"),

relief=FLAT).place(x=230, y=210)

l48 = Label(root7, text="This can make a query from people into action, thus saving multiple lives."

" Everyone has the potential to be an organ, eye and tissue donor. ", bg="black", fg="Grey",

font=("comicsansms", 15, "bold"),

relief=FLAT).place(x=250, y=260)

l49 = Label(root7, text="One person can save and heal up to 60 lives through "

"organ, eye and tissue donation!.", bg="black", fg="Grey",

font=("comicsansms", 15, "bold"),

relief=FLAT).place(x=230, y=310)

l50 = Label(root7, text=" We can find the percentile rate of organ donors per year easily. "

".", bg="black", fg="Grey",

font=("comicsansms", 15, "bold"),

relief=FLAT).place(x=320, y=360)

l50 = Label(root7, text=".", bg="black", fg="Grey",

font=("comicsansms", 15, "bold"),

relief=FLAT).place(x=450, y=410)

btn45 = Button(root7, text="Back", bg="black", fg="grey", command=root7.destroy).place(x=1100, y=620)

root7.mainloop()

#Trigger Add access code window

def Add\_access\_code():

root5 = Tk()

root5.maxsize(400, 400)

root5.minsize(400, 400)

root5.title("ACCESS CODE")

root5.configure(bg="black")

global a, b, c, d, x

a, b, c, d, x = [], [], [], [], ""

def Add(x):

list1.insert(END, "{}: {}".format(nme.get(), x))

try:

f = open("assigned\_name", "x")

t = open("access\_code", "x")

if f:

if t:

f.close()

f = open("assigned\_name", "a")

f.write("{}\n".format(nme.get()))

f.close()

t = open("access\_code", "a")

t.write("{}\n".format(x))

t.close()

listbox2.insert(END, x)

listbox.insert(END, nme.get())

except:

print("already created")

f = open("assigned\_name", "a")

f.write("{}\n".format(nme.get()))

f.close()

t = open("access\_code", "a")

t.write("{}\n".format(x))

t.close()

listbox2.insert(END, x)

listbox.insert(END, nme.get())

#generating random Access Code For user Login

def generate():

global a, b, c, d, x

a, b, c, d, x = [], [], [], [], ""

o = 65

for i in range(o, o + 26):

p = o + 33

for j in range(p - 1, p):

if o < 75:

for g in range(o - 64):

c.append(g)

l = chr(j)

d.append(l)

k = chr(i)

a.append(k)

o += 1

if o == 92:

break

for i in range(2):

n = random.choice(a)

h = random.choice(c)

f = random.choice(d)

b.append(h)

b.append(f)

b.append(n)

for i in range(len(b)):

x = str(x + "{}".format(b[i]))

e1.insert(1, x)

Add(x=x)

f5 = Frame(root5, width="200", bg="grey", borderwidth=2)

l99 = Label(f5, text="Name:").place(x=10, y=1)

nme = Entry(f5, borderwidth=2)

nme.place(x=20, y=30)

e1 = Entry(f5, borderwidth=2)

e1.place(x=20, y=79)

btn9 = Button(f5, text="Generate", command=generate).place(x=20, y=109)

f5.pack(side=RIGHT, fill=Y)

f6 = Frame(root5, width="200", relief=SUNKEN, bg="grey", borderwidth=2)

list1 = Listbox(f6, height="90", width="200")

list1.pack(fill=Y, padx=6, pady=6)

f6.pack(side=LEFT, fill=Y)

root5.mainloop()

#destroying root and calling main

def dec():

root.destroy()

main()

menubar = Menu(root, bg="grey", font=("comicsansms", 12, "bold"))

Set = Menu(menubar, tearoff=0)

Set.add\_command(label="Access Code", font=("comicsansms", 12, "bold"), command=Add\_access\_code)

Set.add\_command(label="Donation form", font=("comicsansms", 12, "bold"), command=item\_values)

Set.add\_separator()

Set.add\_command(label="Exit", font=("comicsansms", 12, "bold"), command=root.destroy)

menubar.add\_cascade(label="Set", menu=Set)

Bill = Menu(menubar, tearoff=0)

Bill.add\_command(label="New Organ Donation Form", font=("comicsansms", 12, "bold"), command=dec)

menubar.add\_cascade(label="Bill", menu=Bill)

Bill.add\_separator()

help = Menu(menubar, tearoff=0)

help.add\_command(label="About", font=("comicsansms", 12, "bold"), command=about)

menubar.add\_cascade(label="Help", menu=help)

root.config(menu=menubar)

f10 = Frame(root, height="10", bg="grey", relief=SUNKEN, borderwidth=5)

l33 = Label(f10, text="List of Access Code", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).pack()

f10.pack(side=TOP, fill=X, padx=10, pady=10)

f12 = Frame(root, height="30", bg="grey")

l34 = Label(f12, text="Assigned To", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(x=50, y=10)

l35 = Label(f12, text="Access Code", bg="grey", fg="black", font=("comicsansms", 12, "bold"),

relief=FLAT).place(x=900, y=10)

f12.pack(side=TOP, fill=X, padx=10)

f11 = Frame(root, height="800", bg="grey", borderwidth=5)

listbox = Listbox(f11, height=900, width=100)

listbox.pack(side=LEFT, fill=Y, padx=6, pady=0)

listbox2 = Listbox(f11, height=900, width=100)

try:

f = open("assigned\_name", "r")

z = f.readlines()

for i in z:

k = i

listbox.insert(1, k)

f.close()

t = open("access\_code", "r")

j = t.readlines()

for i in j:

k = i

listbox2.insert(1, k)

except:

messagebox.showinfo("information", "no Access code")

listbox2.pack(side=RIGHT, fill=Y, padx=6, pady=0)

f11.pack(side=TOP, fill=X, padx=10)

btn = Button(f12, text="Delete Name",relief=FLAT, command=lambda listbox=listbox: listbox.delete(ANCHOR))

btn.place(x=450, y=1)

btn90 = Button(f12, text="Delete Access code",relief=FLAT, command=lambda listbox2=listbox2: listbox2.delete(ANCHOR))

btn90.place(x=1150, y=1)

root.mainloop()

#it will decide on the bases of Entered Access code that, the end user is adminstrator or user

#So, as a result it will call Admin window or User window

def pro():

# f = open("sign\_in.log", "w").write(sie.get())

try:

f = open("pa", "r")

f.close()

except:

messagebox.showinfo("Error","Create a User Account First\n Click on SignUp to create")

k = sie.get()

if k != "":

f = open("pa", "r")

s = f.readline()

if s != k:

y = open("access\_code", "r")

d = y.readline()

if k in d:

messagebox.showinfo("information", "User Login Success")

root1.destroy()

f.close()

main()

else:

f.close()

messagebox.showinfo("warning", "Incorrect Access Code")

else:

dis3()

else:

messagebox.showinfo("Information", "Enter Access Code")

# n = root

root1.title("")

# root1.minsize(965, 460)

# root1.maxsize(965, 460)

f1 = Frame(root1, bg="black", height=460, width=965).pack()

# root1.configure(bg="black")

l35 = Label(f1, text="SignIn TO ORGAN DONATION APP.", bg="black", fg="grey",

font=("comicsansms", 30, "bold")).place(x=190, y=80)

l36 = Label(f1, text="Access Code", bg="black", fg="grey", font=("comicsansms", 20, "bold"), relief=FLAT).place(

x=270, y=170)

global sie

sie = Entry(f1, show="\*", relief=SUNKEN, borderwidth=5)

sie.place(x=450, y=170)

btn8 = Button(f1, text="SignIn", bg="Grey", fg="Black", font=("comicsansms", 20, "bold"),

command=pro).place(x=620, y=300)

l37 = Label(f1, text="If you are new user,then", bg="black", fg="grey",

font=("comicsansms", 20, "bold"),

relief=FLAT).place(x=220, y=360)

btn9 = Button(f1, text="SignUp", bg="black", fg="Blue", font=("comicsansms", 10, "bold"), relief=FLAT,

command=dis).place(x=650, y=360)

root1.mainloop()

#destroy root window then,go for signin window

def dis():

root.destroy()

sgnin()

#destroy root window then,go for signup window

def dis2():

root.destroy()

sgnup()

root.title("")

photo = PhotoImage(file="d.png")

x\_lable = Label(image=photo)

Label.pack(self=x\_lable)

root.minsize(880, 460)

root.maxsize(880, 460)

root.configure(bg="black")

l34 = Label(root, text="\n Welcome to ORGAN DONATION \n Application", bg="black", fg="grey",

font=("comicsansms", 30, "bold")).place(x=130, y=80)

btn6 = Button(root, text="SignIn", bg="Grey", fg="Black", font=("comicsansms", 20, "bold"),

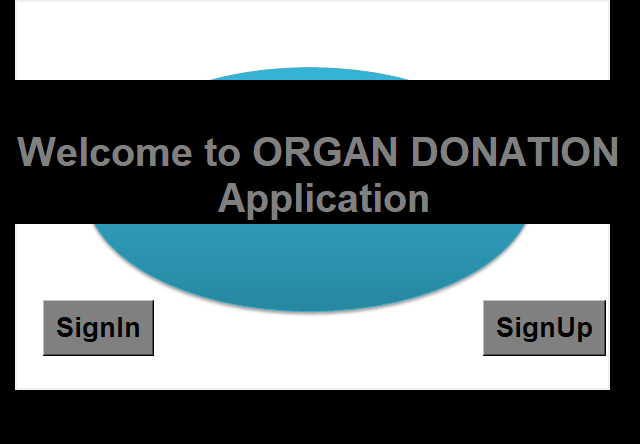
command=dis).place(x=170, y=300)

btn7 = Button(root, text="SignUp", bg="Grey", fg="Black", font=("comicsansms", 20, "bold"),

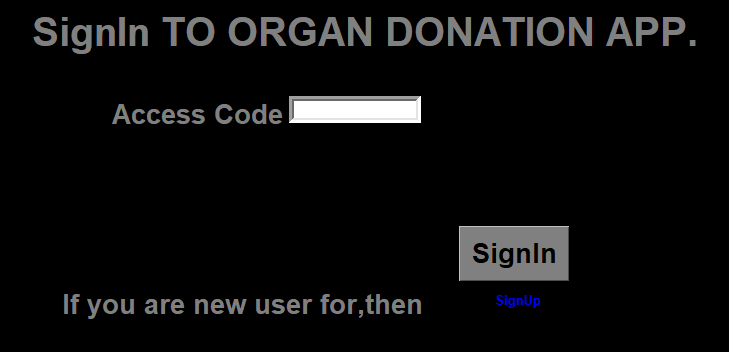
command=dis2).place(x=610, y=300)

root.mainloop()

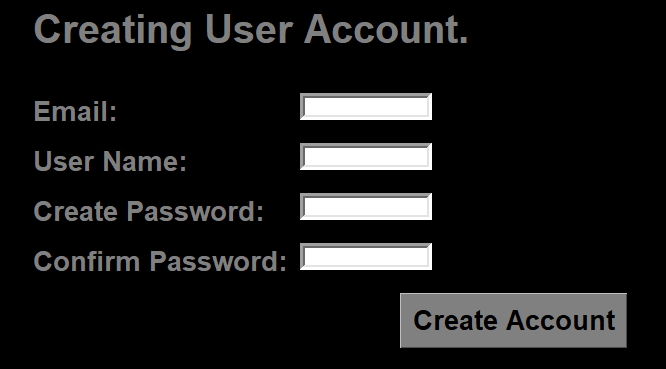
**Sample Screen shots**

****

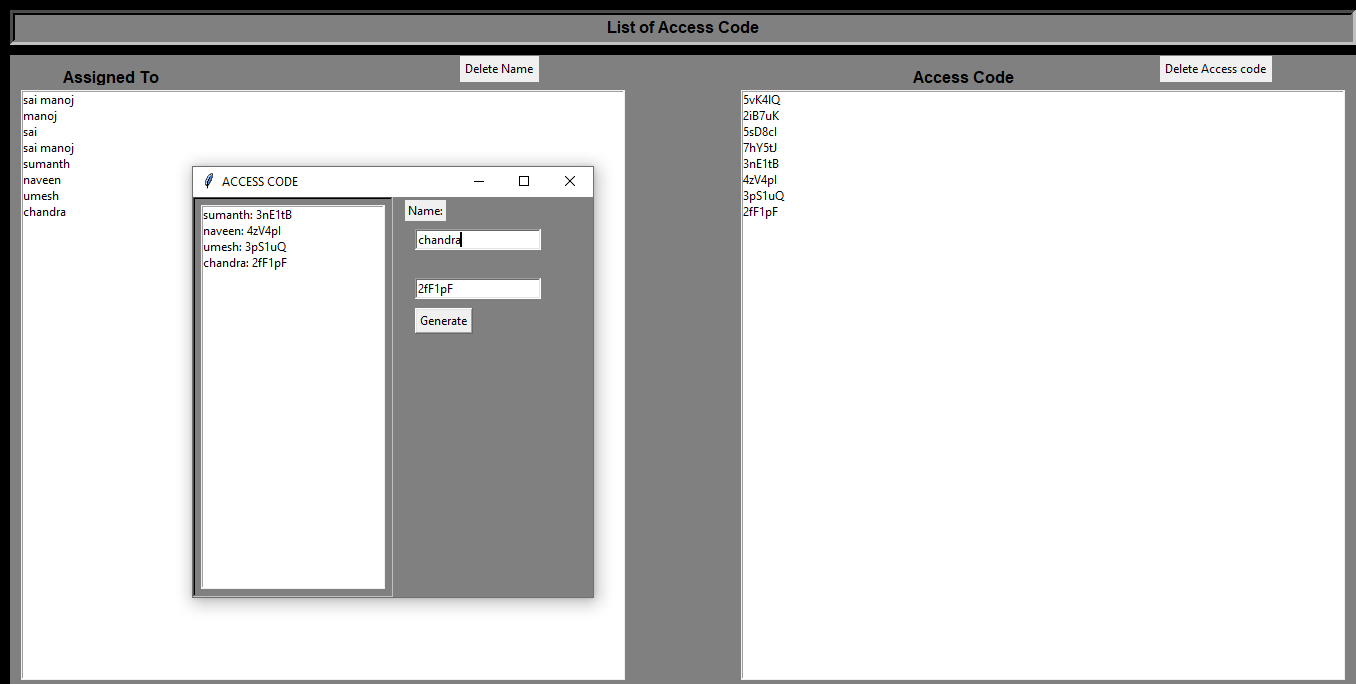
**Figure b.1 Welcome screen and login or signup page**

****

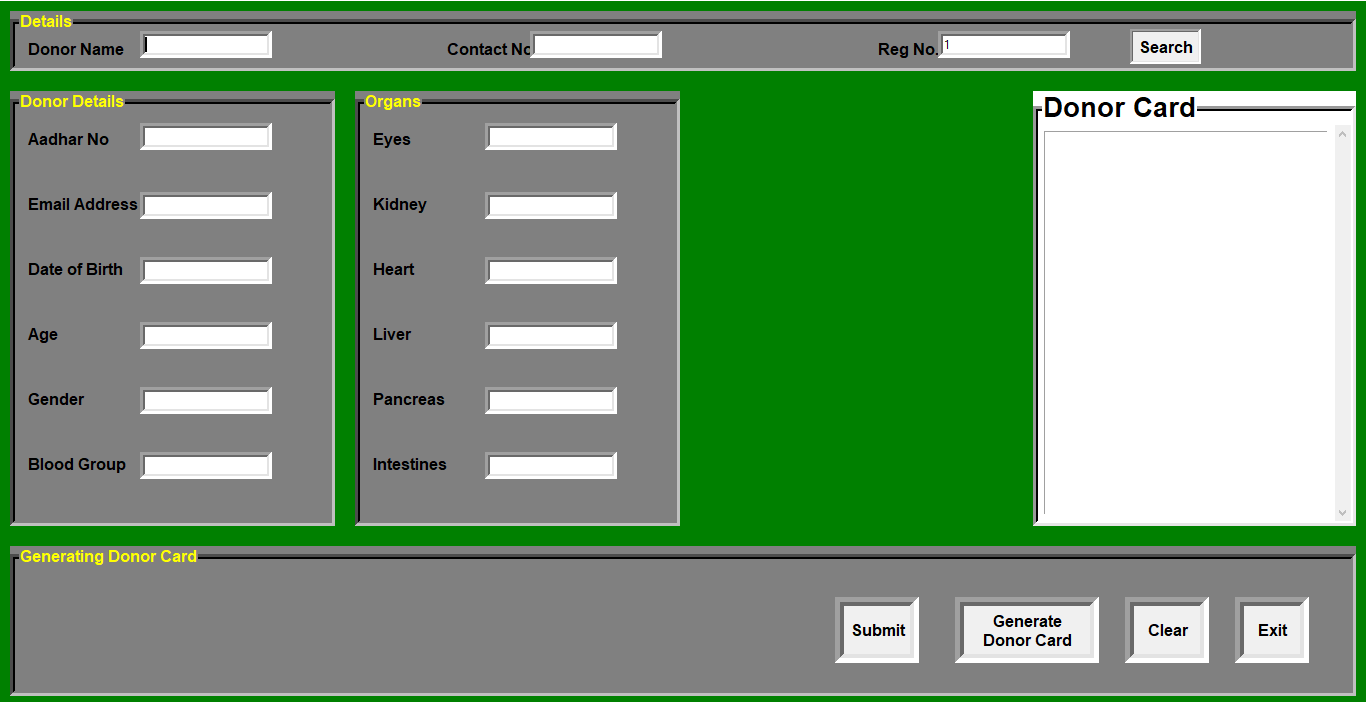
**Figure b.2 User login using access codes**

****

**Figure b.3 Creation of new account**

****

**Figure b.4 Generating access codes for users by admin**

****

**Figure b.5 Organs donation form**