



SUBJECT – COMPUTER SCIENCE



MINDCOLOGY

BOOKING SYSTEM FOR THERAPY AND MENTAL HEALTH INSTITUTIONS

- **By Arya Hariharan, Shamili Gande and Raj Ammu**

CERTIFICATE

This is to certify that ARYA HARIHARAN , Roll No. : _____ of Chrysalis High, Kadugodi has successfully completed Computer Project towards partial completion of Practical Examination of AISSCE 2022 as prescribed by CBSE.

Signature of Internal Examiner

Signature of External Examiner

PRINCIPAL

Table of Contents

Acknowledgment

Abstract

Applications and Tools Used

Systems Software and Hardware Used

Structure of the Project

Project Map

Source Code

Output and Screenshots

Conclusion

Limitations

Bibliography

ACKNOWLEDGEMENT

We would like to sincerely and profusely thank our Computer Science teacher Ms. Sonia Chacko for her guidance and support in completing my project.

We would also like to extend my gratitude to our Principal, Ms. Sailatha Rajesh for providing us with all the facility that was required.

Last but not the least, we would extend our gratitude towards all teaching and non-teaching staff of Chrysalis High, Kadugodi and towards our friends who have supported us to complete this project.

ABSTRACT

The intention behind the computer science project is to create a program-driven system that is unique and one that may be beneficial to the common people when thought over on a large scale. In this era of the pandemic, access to therapy and aid for mental health issues has been limited and the main intention of this project is to help curb that problem.

This project is essentially a booking system by which a user can book sessions with associated therapy institutions and connect with support groups. The information available to the user will include doctor's qualification, the price for consultations, specialization and so on. A price will be calculated based on a fixed formula and will be displayed to the user as the final output in the form of an invoice.

WHY DID WE CHOOSE THIS PROJECT?

There are several online appointment booking systems in the market. The concept of having a booking system associated only with mental health institutions and therapy centres is still upcoming. However, the importance of such institutions is growing readily over time, and we chose this project as perhaps a stepping-stone towards a larger idea through which we can improve the reach of such institutions, making them more accessible to anyone and everyone. The concept is focused on reducing the stigma around asking for help regarding mental health and making it easier for people seeking help to avail it.

PROJECT COMPONENTS/FEATURES

The project has the following major components/features within it –

1. **CHOICE OF DOCTORS AND INSTITUTIONS** - This feature allows one to choose between a variety of doctors and institutions, all specializing in therapy and mental health recovery. They range from general therapists to specialized doctors. A user may also choose from several associated workshops and support groups, if they feel they'd rather prefer a group experience by interacting with others who are going through the same issues.

2. USER PREFERENCES – Every user will automatically be added to the mailing list upon registration to receive further notifications. However, if a user wishes to unsubscribe, the feature for that has been provided.

3. SIMPLE CALCULATION OF BILL AMOUNT - Once the user has finalised which doctor/institution/workshop they would like to consult with, they shall be directed to a confirmation page where they can choose the date and the doctor. Based on the individual price for each doctor, a simple calculation incorporating GST and convenience fee is done to display the final price to the user on the invoice. The invoice contains other details like invoice number, doctor's name, and date of appointment.

APPLICATIONS AND TOOLS USED

1. PYTHON

Python is a high-level general-purpose programming language. It is an interpreted language that emphasises on code readability with high indentation. It is an object-oriented language that help programmers right clear and logical code for both small and large project.

The major features of Python are –

- (i) It is an interpreted language. Hence debugging is easier.
- (ii) It is platform independent and hence can be used on any device and is compatible with all major operating systems.
- (iii) It is completely free of cost to download. It is also an open-source software.
- (iv) It is very simple to use since the language resembles the English language closely. Hence, it is very easy to learn how to write the command and expressions.
- (v) It also has a rich library support such as Python Standard Library that provides a variety of functions for use.

Python has become extremely popular as a programming language due to its ease of use and understanding. It is a simple language that is easy to learn and implement to create a variety of different applications. The inbuilt libraries provide a variety of functions for ready use, further increasing the ease of use.

2. STRUCUTRED QUERY LANGUAGE

SQL (Structured Query Language) is a standardised language developed and used for accessing and modifying relational databases. The uses of SQL include modifying database table and index structures; adding, updating and deleting rows of data; and retrieving subsets of information from within a database.

Its unique features/characteristics are –

- (i) It can be installed in various devices and is supported by a variety of operating systems.
- (ii) It is highly secure with password protection and also has the feature of providing excellent connectivity. SQL can be used as a back-end feature for projects and such connectivity can be established by a variety of programming languages.
- (iii) It is also easy to use since it is a queried language.

SYSTEM SOFTWARE AND HARDWARE USED

SOFTWARE USED TO RUN THE PROJECT-

- 1) Windows 10.0 and iOS
- 2) Python IDLE and Visual Studio Code
- 3) Front end
 - Tkinter
 - Python
- 4). Back end
 - SQL (Structured Query Language)

HARDWARE USED TO RUN THE PROJECT-

- 1) 1.10 GHz Dual-Core Processor
- 2) 4/8 GB RAM
- 3) 250 GB HDD

STRUCTURE OF THE PROJECT

The main idea behind this project is to provide a user with the choice of choosing from a variety of different consultation methods and doctors to increase comfort and to cater to the user's needs. For this purpose, we have categorised the information displayed in the project into four broad categories - welcome pages, general doctors, specialist doctors and workshops. These are the four main segments or 'features' of our project, in which the code has been written in a single .py file. A brief idea of each segment is given below -

WELCOME PAGES

These are the first set of pages that are a part of our application. This set includes the home page, login page, register page, welcome page, and available services page. The first page of our application is the home page which displays the name, motto, and login/register buttons. On clicking on these buttons, the user is directed towards the login and register page respectively. This is the segment of the project where there is a core utilisation of SQL. If it is a new user, the data from the register page (name, username, password, and email) is split and stored in two tables - one for user info (name, username, and password) and the other for mailing list info (email, name). The information from these tables is then accessed when an existing user tries to login by entering their username and password. Only if both the username and password are correct and matching are they allowed to proceed. Else, an error message is displayed. Once the user has successfully registered or logged in, they are taken to the 'Welcome Page' where a brief description of the application is written. This page also has an 'About Us' button at the bottom which takes the user to another page where we provide a brief introduction about ourselves. Then, users can navigate to the 'Available Services Page' where we display all the services available - consultations with general doctors, specialists, and workshops.

GENERAL DOCTORS PAGES

These pages contain information regarding the general doctors that are available for contact through our program. These doctors are for those users who feel like their issue is not specific or localised on some feeling and would just like to have a general talk. The main

segmentation begins from the page which we call the 'Available Services Page' where a user can navigate to the list of general doctors by clicking a button. On the general doctors' page, we have listed out the name, qualification and associated institution. One can click on the doctor of their choice and navigate to a 'Confirmation Page' where they must reconfirm their doctors via radio buttons and choose a date from a calendar implemented using the tkCalender module. Once done, they are directed to a final invoice page and then to a logout page where they can choose to exit the application. Each button in this page, as well on most other pages, is an image converted to a button and then added on to the main page to give the user a broader area on which to click to take them to their desired page. We have also enabled a feature where hovering over an image-button will cause its border width to change from 0 to 1, acting as further indication that the user is over a button and is free to click on it. On the invoice page, the final price after calculations will also be shown. [PRICE+GST+CONVENIENCE FEE]

SPECIALISTS PAGES

These pages contain information regarding the specialist doctors that are available for contact through our program. These doctors are for those users who feel like their issue is specific and would like to contact a doctor whose expertise lies in helping curb that specific issue. There are a total of eight doctors displayed - two doctors for each specialisation i.e., anxiety, depression, post trauma or addiction. The main segmentation begins from the page which we call the 'Available Services Page' where a user can navigate to the list of specialist doctors by clicking a button. On the specialists' page, we have listed out the name, qualification, and associated institution. One can click on the doctor of their choice and navigate to a 'Confirmation Page' where they must reconfirm their doctors via radio buttons and choose a date from a calendar implemented using the tkCalender module. Once done, they are directed to a final invoice page and then to a logout page where they can choose to exit the application. Each button in this page, like the other pages, is an image converted to a button and then added on to the main page to give the user a broader area on which to click to take them to their desired page. We have also enabled a feature where hovering over an image-button will cause its border width to change from 0 to 1, acting as further indication that the user is over a button and is free to click on it. On the

invoice page, the final price after calculations will also be shown.
[PRICE+GST+CONVENIENCE FEE]

WORKSHOP PAGES

These pages contain information regarding the workshops/support groups that are available for contact through our program. These are for those users who feel like their issue is helped better by identifying people like themselves around them and moving forward together towards healing as a group. There are a total of eight workshops displayed - two workshops for each specialisation i.e., anxiety, depression, post-trauma, or addiction. The main segmentation begins from the page which we call the 'Available Services Page' where a user can navigate to the list of workshops by clicking a button. On the workshops page, we have listed out the name, brief description, and associated institution of the workshop. One can click on the workshop of their choice and navigate to a 'Confirmation Page' where they must reconfirm their workshop via radio buttons and choose a date from a calendar implemented using the tkCalender module. Once done, they are directed to a final invoice page and then to a logout page where they can choose to exit the application. Each button in this page, like the other pages, is an image converted to a button and then added on to the main page to give the user a broader area on which to click to take them to their desired page. We have also enabled a feature where hovering over an image-button will cause its border width to change from 0 to 1, acting as further indication that the user is over a button and is free to click on it. On the invoice page, the final price after calculations will also be shown.

Apart from these groups, we also have some other stand-alone pages like the 'Terms and Conditions Page' and 'Unsubscribe from Mailing List Page' which can be accessed by clicking on the user logo displayed on every single page on the top-right corner. The 'Unsubscribe from Mailing List Page' also accesses SQL to delete the mailing info of a user who no longer wishes to receive emails from Mindcology.

INBUILT LIBRARIES AND PACKAGES USED

TKINTER: Tkinter is Python's de-facto standard GUI (Graphical User Interface) package. GUI is nothing but a desktop app that provides you with an interface that helps you to interact with the computers and enriches your experience of giving a command to your code.

METHODS USED:

a. Geometry(): This method is used to set the dimensions of the Tkinter window as well as it is used to set the position of the main window on the user's desktop.

b. Frame(): It works like a container, which is responsible for arranging the position of other widgets.

c. Grid(): This geometry manager organizes widgets in a table-like structure of the parent widget.

d. Pack(): This geometry manager organizes widgets in blocks before placing them in the parent widget.

e. place(): This geometry manager organizes widgets by positioning them in a specific position, in the parent widget.

f. Label(): This widget implements a display box where you can place text or images. The text displayed by this widget can be updated at any time you want.

g. Button(): The Button widget is used to add Buttons in Python. These buttons can display text or images that convey the purpose of the buttons. You can attach a function or a method to a button which is called automatically when you click the button.

h. Calendar(): This widget implements a calendar on a desired page from which one can extract the chosen date using functions.

i. Canvas(): This method is used to create a canvas which is a rectangular area intended for drawing pictures or other complex layouts. We can place graphics, text, widgets, or frames in a canvas.

j. Label(): This widget implements a display box where you can place text or images. The text displayed by this widget can be updated at any point of time.

k. Button(): This widget is used to add Button in a Python application. These buttons. These buttons can display text or images to convey their purpose. We can attach a function or method to a button which is executed once the button is clicked.

l. Radiobutton(): This widget implements a multiple-choice button, which is a way to offer many possible selections to the user but lets them choose only one of them.

m. Entry(): This widget is used to accept single-line strings from the user.

n. Scrollbar(): This widget is used to scroll down content. It is possible to create both horizontally and vertically oriented scrollbars.

o. bind(): This function is used to deal with events. It is possible to bind functions and methods to an event and in turn, bind these functions and methods to any widget.

We also used various methods from the Python-MySQL interface to establish the connection between the program and the database and perform operations on it –

a. **connect():** Establishes a connection to the MySQL database.

b. **cursor():** Creates a cursor object. This is the object you use to interact with the database.

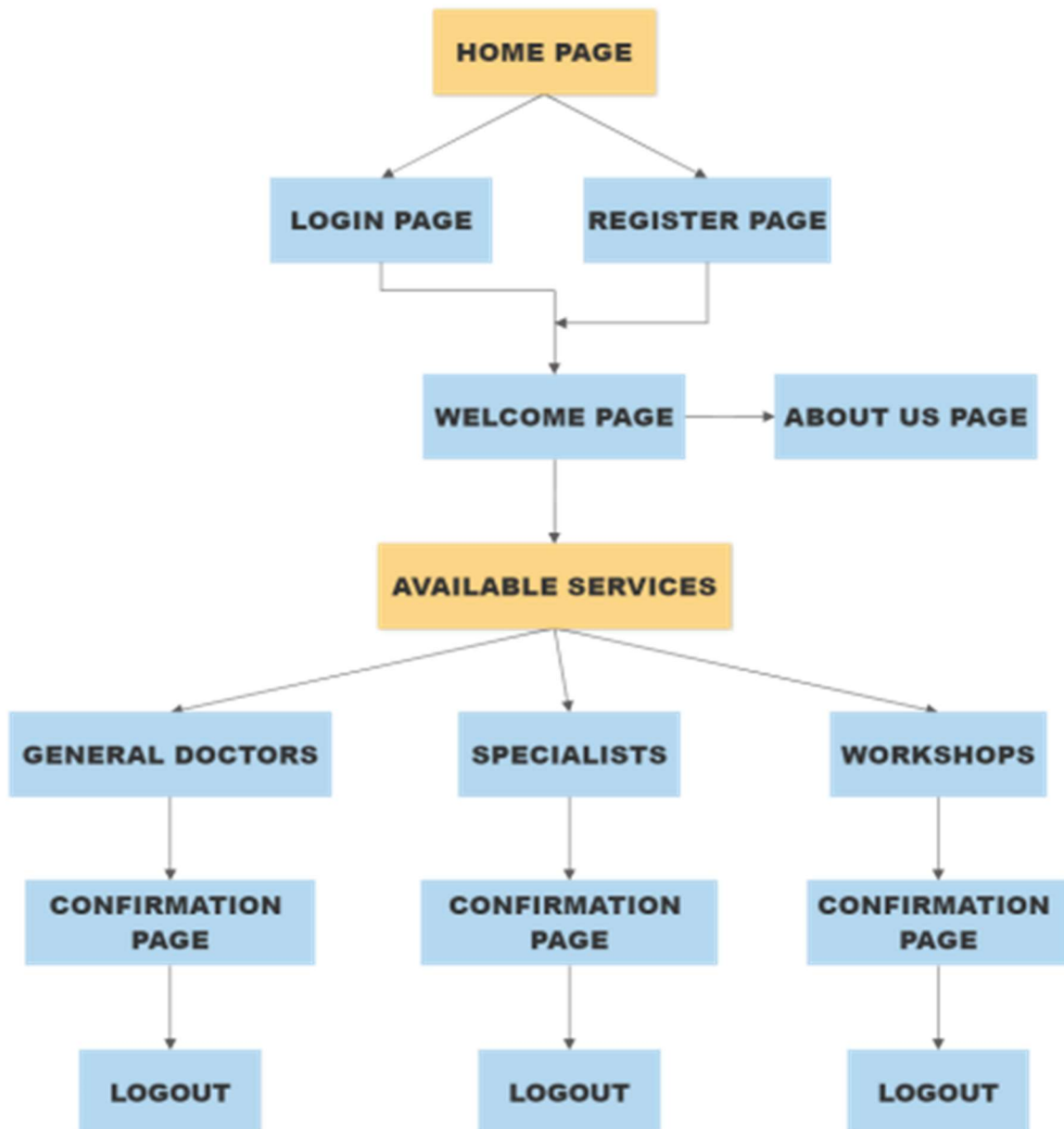
c. **execute():** This method executes the given database operation.

d. **fetchall():** The fetchall() method retrieves all rows of a query result, returning them as a sequence of sequences.

e. **commit():** This method commits the current transaction.

f. **rollback():** This method reverts the changes made by the current transaction.

PROJECT REPORT



***NOTE - ALL THE LOGOUT PAGES AREN'T DIFFERENT. EACH CONFIRMATION PAGE LEADS TO THE SAME LOGOUT PAGE**

Also, apart from the flow given above, the user can also navigate to the 'Terms and Conditions Page' or the 'Unsubscribe from Mailing List Page'.

SOURCE CODE

```
#Main Page Code

from tkinter import *

import tkinter as tk

from PIL import ImageTk, Image

import mysql.connector

from tkinter import messagebox

from tkinter import ttk

from tkcalendar import *

from functools import partial

import random


def endProject():

    MPage.destroy()


fullName=' '


def nameForInvoice(firstname,lastname):

    global a

    a=firstname

    global fullName
```

```
fullName=firstname + ' ' + lastname
```

```
def unsubscribe():
```

```
    def endSubscription():
```

```
        connection = mysql.connector.connect(
```

```
            database='mindcology',
```

```
            user='root',
```

```
            password='12ammu34')
```

```
        cursor=connection.cursor()
```

```
        instruction = "DELETE FROM MAILING_LIST WHERE FIRSTNAME = '%s' " % (a)
```

```
        try:
```

```
            cursor.execute(instruction)
```

```
            connection.commit()
```

```
        except:
```

```
            conn.rollback()
```

```
            messagebox.showinfo('Error','Entry Already Deleted')
```

```
    U1Page=Toplevel(MPage)
```

```
    background=Canvas(U1Page,width=600,height=600)
```

```
    image=ImageTk.PhotoImage(Image.open("Final Unsubscribe Page.jpg"))
```

```
    label1=Label(image=image)
```



```

label1.image=image

background.create_image(0,0,anchor='nw',image=image)

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

UPage.geometry('600x600')

UPage=Toplevel(MPage)

background=Canvas(UPage,width=600,height=600)

image=ImageTk.PhotoImage(Image.open("Unsubscribe Page.jpg"))

label1=Label(image=image)

label1.image=image

background.create_image(0,0,anchor='nw',image=image)

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

UPage.geometry('600x600')

unsubscribeButton=Button(UPage,text="CONFIRM",font=('Bahnschrift
Condensed',18),bg='white',fg='red',command=endSubscription, borderwidth=1,relief="solid")

unsubscribeButton.place(x=470,y=395)

#Terms and Conditions Page

def TandC():

    TCPage=Toplevel(MPage)

    background=Canvas(TCPage,width=600,height=600)

```

```
image=ImageTk.PhotoImage(Image.open("Terms and Conditions Page.jpg"))
```

```
label1=Label(image=image)
```

```
label1.image=image
```

```
background.create_image(0,0,anchor='nw',image=image)
```

```
background.pack(expand=True,fill=BOTH)
```

```
TCPage.geometry('600x600')
```

```
#Logout Page
```

```
def logout():
```

```
    LOPage=Toplevel(MPage)
```

```
    background=Canvas(LOPage,width=600,height=600)
```

```
    image=ImageTk.PhotoImage(Image.open("Logout Page.jpg"))
```

```
    label1=Label(image=image)
```

```
    label1.image=image
```

```
    background.create_image(0,0,anchor='nw',image=image)
```

```
    background.pack(expand=True,fill=BOTH)
```

```
    LOPage.geometry('600x600')
```

```
    decisionButton=Button(LOPage,text="YES - CONFIRM LOGOUT",font=('Bahnschrift
```

```
Condensed',18),bg='white',fg='red',command=endProject, borderwidth=1,relief="solid")
```

```
    decisionButton.place(x=270,y=510)
```

```
#Workshop Confirmation Page
```

```
def workshopConfirmation():
```

```
    def workshopInvoice(recievedValue):
```

```
        Name=fullName.title()
```

```
        def grab_date():
```

```
            date=cal.get_date()
```

```
            return date
```

```
        date=grab_date()
```

```
        listForDate=date.split('/')
```

```
        finalDate=listForDate[1]+'-'+listForDate[0]+'-'+listForDate[2]
```

```
        price="₹"+str((1499*(2/100))+1499+((10/100)*1499))
```

```
        WIPage=Toplevel(MPage)
```

```
        background=Canvas(WIPage,width=600,height=600)
```

```
        image=ImageTk.PhotoImage(Image.open("Invoice Page.jpg"))
```

```
        label1=Label(image=image)
```

```
        label1.image=image
```

```
        background.create_image(0,0,anchor='nw',image=image)
```

```
        background.pack(expand=True,fill=BOTH)
```

```
        WIPage.geometry('600x600')
```

```
nameLabel=Label(WIPage, text=Name, font=('Bahnschrift Condensed',22),bg='black',fg='gold')

nameLabel.place(x=350,y=180)

workshopLabel=Label(WIPage,text=recievedValue,font=('Bahnschrift
Condensed',22),bg='black',fg='gold')

workshopLabel.place(x=325,y=262)

priceLabel=Label(WIPage,text=price,font=('Bahnschrift Condensed',22),bg='black',fg='gold')

priceLabel.place(x=375,y=342)

dateLabel=Label(WIPage,text=finalDate,font=('Bahnschrift Condensed',22),bg='black',fg='gold')

dateLabel.place(x=375,y=424)

done=Button(WIPage, text="Done",font=('Bahnschrift
Condensed',18),bg='black',fg='gold',command=logout)

done.place(x=265, y=478)

invoiceNumber=Label(WIPage,text='Invoice No. :
'+str(random.randint(0,10000000)),font=('Bahnschrift Condensed',16),bg='black',fg='gold')

invoiceNumber.place(x=420, y=20)

def selected():

    a=var.get()

    submit=ImageTk.PhotoImage(Image.open("Submit Button.jpg"))

    submitButtonLabel=Label(image=submit)

    submitButtonLabel.image=submit
```

```
submitButton=Button(WCPage,image=submit,borderwidth=0,highlightthickness=0,command=partial(
workshopInvoice,a))

    submitButton.place(x=230,y=620)

WCPage=Toplevel(MPage)

background=Canvas(WCPage,width=600,height=700)

image=ImageTk.PhotoImage(Image.open("Confirmation Page.jpg"))

label1=Label(image=image)

label1.image=image

background.create_image(0,0,anchor='nw',image=image)

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

WCPage.geometry('600x700')

cal=Calendar(WCPage,selectmode="day",year=2022,month=1,day=22)

cal.place(x=25,y=310)

var=tk.StringVar()

ade=Radiobutton(WCPage,text='Addiction Extinction - Rs1499/session',font=('Bahnschrift
Condensed',14),variable=var,value='Addiction
Extinction',command=selected,tristatevalue=0,bg="black",fg="gold")

ade.place(x=307,y=280)
```

```
aa=Radiobutton(WCPage,text='Absolut Anxiety - Rs1499/session',font=('Bahnschrift
Condensed',14),variable=var,value='Absolut
Anxiety',command=selected,tristatevalue=0,bg="black",fg="gold")

aa.place(x=307,y=330)

at=Radiobutton(WCPage,text='Against Trauma- Rs1499/session',font=('Bahnschrift
Condensed',14),variable=var,value='Against
Trauma',command=selected,tristatevalue=0,bg="black",fg="gold")

at.place(x=307,y=380)

dod=Radiobutton(WCPage,text='Dawn Over Depression- Rs1499/session',font=('Bahnschrift
Condensed',14),variable=var,value='Dawn Over
Depression',command=selected,tristatevalue=0,bg="black",fg="gold")

dod.place(x=307,y=430)

fah=Radiobutton(WCPage,text='Feeling and Healing - Rs1499/session',font=('Bahnschrift
Condensed',14),variable=var,value='Feeling and
Healing',command=selected,tristatevalue=0,bg="black",fg="gold")

fah.place(x=307,y=480)

ada=Radiobutton(WCPage,text='Adios Addiction - Rs1499/session',font=('Bahnschrift
Condensed',14),variable=var,value='Adios
Addiction',command=selected,tristatevalue=0,bg="black",fg="gold")

ada.place(x=307,y=530)
```

```
#Workshop Page
```

```
def workshop():
```

```
    WPage=Toplevel(MPage)
```

```
    main_frame=Frame(WPage)
```

```
    main_frame.pack(fill=BOTH, expand=1)
```

```
    my_canvas = Canvas(main_frame)
```

```
    my_canvas.pack(side=LEFT, fill=BOTH, expand=1)
```

```
    my_scrollbar = ttk.Scrollbar(main_frame, orient=VERTICAL, command=my_canvas.yview)
```

```
    my_scrollbar.pack(side=RIGHT, fill=Y)
```

```
    my_canvas.configure(yscrollcommand=my_scrollbar.set)
```

```
    my_canvas.bind('<Configure>', lambda e: my_canvas.configure(scrollregion =  
my_canvas.bbox("all")))
```

```
    second_frame = Frame(my_canvas)
```

```
    my_canvas.create_window((0,0), window=second_frame, anchor="nw")
```

```
    background=Canvas(second_frame,width=600,height=1445)
```

```
    image=ImageTk.PhotoImage(Image.open("Workshops Page.jpg"))
```

```
    label1=Label(image=image)
```

```
    label1.image=image
```

```
background.create_image(0,0,anchor='nw',image=image)
```

```
background.pack(expand=True,fill=BOTH)
```

```
WPage.geometry('600x1000')
```

```
def item_selected(event):
```

```
    selected_index = userOptions.curselection()
```

```
    for i in selected_index:
```

```
        if userOptions.get(i)=='Unsubscribe':
```

```
            unsubscribe()
```

```
        if userOptions.get(i)=='Logout':
```

```
            logout()
```

```
        if userOptions.get(i)=='T & C':
```

```
            TandC()
```

```
def button_hover1(a):
```

```
    addictionExtinctionButton["borderwidth"]=1
```

```
def button_hover_leave1(b):
```

```
    addictionExtinctionButton["borderwidth"]=0
```

```
def button_hover2(a):
```

```
    absolutAnxietyButton["borderwidth"]=1
```

```
def button_hover_leave2(b):
```

```
    absolutAnxietyButton["borderwidth"]=0
```



```
def button_hover3(a):

    dawnOverDepressionButton["borderwidth"]=1

def button_hover_leave3(b):

    dawnOverDepressionButton["borderwidth"]=0

def button_hover4(a):

    againstTraumaButton["borderwidth"]=1

def button_hover_leave4(b):

    againstTraumaButton["borderwidth"]=0

def button_hover5(a):

    adiosAddictionButton["borderwidth"]=1

def button_hover_leave5(b):

    adiosAddictionButton["borderwidth"]=0

def button_hover6(a):

    feelingAndHealingButton["borderwidth"]=1

def button_hover_leave6(b):

    feelingAndHealingButton["borderwidth"]=0

def button_hover7(a):

    global userOptions

    options= ('Unsubscribe','T & C','Logout')

    options_var = tk.StringVar(value=options)
```

```
userOptions= tk.Listbox(second_frame,listvariable=options_var,height=3,selectmode='browse')

userOptions.place(x=450,y=50)

userOptions.bind('<<ListboxSelect>>',item_selected)

def button_hover_leave7(b):

    def button_hover8(a):

        userOptions['borderwidth']=1

    def button_hover_leave8(b):

        userOptions.destroy()

    userOptions.bind("<Enter>",button_hover8)

    userOptions.bind("<Leave>",button_hover_leave8)

back=ImageTk.PhotoImage(Image.open("Back-Button-Logo.jpg"))

backButtonLabel=Label(image=back)

backButtonLabel.image=back

backButton=Button(second_frame,image=back,borderwidth=0,highlightthickness=0,command=servic
es)

backButton.place(x=450,y=20)

user=ImageTk.PhotoImage(Image.open("User Logo.jpg"))

userButtonLabel=Label(image=user)

userButtonLabel.image=user
```

```
userButton=Button(second_frame,image=user,borderwidth=0,highlightthickness=0)
```

```
userButton.place(x=540,y=12)
```

```
addictionExtinction=ImageTk.PhotoImage(Image.open("Addiction Extinction.jpg"))
```

```
addictionExtinctionButtonLabel=Label(image=addictionExtinction)
```

```
addictionExtinctionButtonLabel.image=addictionExtinction
```

```
addictionExtinctionButton=Button(second_frame,image=addictionExtinction,command=workshopConfirmation,borderwidth=0,highlightthickness=0)
```

```
addictionExtinctionButton.place(x=0,y=138)
```

```
absolutAnxiety=ImageTk.PhotoImage(Image.open("Absolut Anxiety.jpg"))
```

```
absolutAnxietyButtonLabel=Label(image=absolutAnxiety)
```

```
absolutAnxietyButtonLabel.image=absolutAnxiety
```

```
absolutAnxietyButton=Button(second_frame,image=absolutAnxiety,command=workshopConfirmation,borderwidth=0,highlightthickness=0)
```

```
absolutAnxietyButton.place(x=0,y=323)
```

```
dawnOverDepression=ImageTk.PhotoImage(Image.open("Dawn Over Depression.jpg"))
```

```
dawnOverDepressionButtonLabel=Label(image=dawnOverDepression)
```

```
dawnOverDepressionButtonLabel.image=dawnOverDepression
```

```
dawnOverDepressionButton=Button(second_frame,image=dawnOverDepression,command=workshopConfirmation,borderwidth=0,highlightthickness=0)
```

```
dawnOverDepressionButton.place(x=0,y=510)
```

```
againstTrauma=ImageTk.PhotoImage(Image.open("Against Trauma.jpg"))
```

```
againstTraumaButtonLabel=Label(image=againstTrauma)
```

```
againstTraumaButtonLabel.image=againstTrauma
```

```
againstTraumaButton=Button(second_frame,image=againstTrauma,command=workshopConfirmation,borderwidth=0,highlightthickness=0)
```

```
againstTraumaButton.place(x=0,y=696)
```

```
adiosAddiction=ImageTk.PhotoImage(Image.open("Adios Addiction.jpg"))
```

```
adiosAddictionButtonLabel=Label(image=adiosAddiction)
```

```
adiosAddictionButtonLabel.image=adiosAddiction
```

```
adiosAddictionButton=Button(second_frame,image=adiosAddiction,command=workshopConfirmation,borderwidth=0,highlightthickness=0)
```

```
adiosAddictionButton.place(x=0,y=879)
```

```
feelingAndHealing=ImageTk.PhotoImage(Image.open("Feeling and Healing.jpg"))
```

```
feelingAndHealingButtonLabel=Label(image=feelingAndHealing)
```

```
feelingAndHealingButtonLabel.image=feelingAndHealing
```

```
feelingAndHealingButton=Button(second_frame,image=feelingAndHealing,command=workshopConfirmation,borderwidth=0,highlightthickness=0)
```

```
feelingAndHealingButton.place(x=0,y=1064)
```

```
addictionExtinctionButton.bind("<Enter>",button_hover1)
```

```
addictionExtinctionButton.bind("<Leave>",button_hover_leave1)
```

```
absolutAnxietyButton.bind("<Enter>",button_hover2)
```

```
absolutAnxietyButton.bind("<Leave>",button_hover_leave2)
```

```
dawnOverDepressionButton.bind("<Enter>",button_hover3)
```

```
dawnOverDepressionButton.bind("<Leave>",button_hover_leave3)
```

```
againstTraumaButton.bind("<Enter>",button_hover4)
```

```
againstTraumaButton.bind("<Leave>",button_hover_leave4)
```

```
adiosAddictionButton.bind("<Enter>",button_hover5)
```

```
adiosAddictionButton.bind("<Leave>",button_hover_leave5)
```

```
feelingAndHealingButton.bind("<Enter>",button_hover6)
```

```
feelingAndHealingButton.bind("<Leave>",button_hover_leave6)
```

```
userButton.bind("<Enter>",button_hover7)
```

```
userButton.bind("<Leave>",button_hover_leave7)
```

```
#Special Doctors Confirmation Page
```

```
def specialConfirmation():
```

```
    def specialInvoice(recievedValue):
```

```
        Name=fullName.title()
```

```
        a=recievedValue.split(", ₹")
```

```
        cost=int(a[1])
```

```
        def grab_date():
```

```
            date=cal.get_date()
```

```
            return date
```

```
        date=grab_date()
```

```
        listForDate=date.split('/')
```

```
        finalDate=listForDate[1]+'-'+listForDate[0]+'-'+listForDate[2]
```

```
        price="₹"+str((cost*(2/100))+cost+((10/100)*cost))
```

```
        SIPage=Toplevel(MPage)
```

```
        background=Canvas(SIPage,width=600,height=600)
```

```
        image=ImageTk.PhotoImage(Image.open("Invoice Page.jpg"))
```

```
        label1=Label(image=image)
```

```
        label1.image=image
```

```
        background.create_image(0,0,anchor='nw',image=image)
```

```
        background.pack(expand=True,fill=BOTH)
```

```

SIPage.geometry('600x600')

nameLabel=Label(SIPage, text=Name,width=21,font=('Bahnschrift
Condensed',20),bg='black',fg='gold')

nameLabel.place(x=305,y=180)

specialLabel=Label(SIPage,text=recievedValue,width=21,font=('Bahnschrift
Condensed',20),bg='black',fg='gold')

specialLabel.place(x=305,y=262)

priceLabel=Label(SIPage,text=price,width=21,font=('Bahnschrift
Condensed',20),bg='black',fg='gold')

priceLabel.place(x=305,y=342)

dateLabel=Label(SIPage,text=finalDate,font=('Bahnschrift Condensed',22),bg='black',fg='gold')

dateLabel.place(x=375,y=424)

done=Button(SIPage, text="Done",font=('Bahnschrift
Condensed',18),bg='black',fg='gold',command=logout)

done.place(x=265, y=480)

invoiceNumber=Label(SIPage,text='Invoice No. :
'+str(random.randint(0,10000000)),font=('Bahnschrift Condensed',16),bg='black',fg='gold')

invoiceNumber.place(x=420, y=20)

def selected():

a=var.get()

```

```
submit=ImageTk.PhotoImage(Image.open("Submit Button.jpg"))
```

```
submitButtonLabel=Label(image=submit)
```

```
submitButtonLabel.image=submit
```

```
submitButton=Button(SCPage,image=submit,borderwidth=0,highlightthickness=0,command=partial(specialInvoice,a))
```

```
submitButton.place(x=230,y=620)
```

```
SCPage=Toplevel(MPage)
```

```
background=Canvas(SCPage,width=600,height=700)
```

```
image=ImageTk.PhotoImage(Image.open("Confirmation Page.jpg"))
```

```
label1=Label(image=image)
```

```
label1.image=image
```

```
background.create_image(0,0,anchor='nw',image=image)
```

```
background.pack(expand=True,fill=BOTH)
```

```
SCPage.geometry('600x700')
```

```
cal=Calendar(SCPage,selectmode="day",year=2022,month=8,day=22)
```

```
cal.place(x=25,y=310)
```

```
var=tk.StringVar()
```



```

dm=Radiobutton(SCPage,text='Diya More-Rs2199/session',font=('Bahnschrift
Condensed',16),variable=var,value='Diya More,
₹2199',command=selected,tristatevalue=0,bg="black",fg="gold")

dm.place(x=315,y=280)

tm=Radiobutton(SCPage,text='Tarun Matthew-Rs2499/session',font=('Bahnschrift
Condensed',16),variable=var,value='Tarun Matthew,
₹2499',command=selected,tristatevalue=0,bg="black",fg="gold")

tm.place(x=315,y=330)

kl=Radiobutton(SCPage,text='Dr. Karim Lok-Rs2299/session',font=('Bahnschrift
Condensed',16),variable=var,value='Dr. Karim Lok,
₹2299',command=selected,tristatevalue=0,bg="black",fg="gold")

kl.place(x=315,y=380)

vg=Radiobutton(SCPage,text='Vijay Ganeshan-Rs2399/session',font=('Bahnschrift
Condensed',16),variable=var,value='Vijay Ganeshan,
₹2399',command=selected,tristatevalue=0,bg="black",fg="gold")

vg.place(x=315,y=430)

ab=Radiobutton(SCPage,text='Ananya Bahri-Rs1999/session',font=('Bahnschrift
Condensed',16),variable=var,value='Ananya Bahri,
₹1999',command=selected,tristatevalue=0,bg="black",fg="gold")

ab.place(x=315,y=480)

```

```

rr=Radiobutton(SCPage,text='Radhika Ramanathan-Rs2399/session',font=('Bahnschrift
Condensed',15),variable=var,value='Radhika Ramanathan,
₹2399',command=selected,tristatevalue=0,bg="black",fg="gold")

rr.place(x=313,y=530)

#Special Doctors Page

def specialdoctors():

    SDPage=Toplevel(MPage)

    main_frame=Frame(SDPage)

    main_frame.pack(fill=BOTH, expand=1)

    my_canvas = Canvas(main_frame)

    my_canvas.pack(side=LEFT, fill=BOTH, expand=1)

    my_scrollbar = ttk.Scrollbar(main_frame, orient=VERTICAL, command=my_canvas.yview)

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

    my_canvas.configure(yscrollcommand=my_scrollbar.set)

    my_canvas.bind('<Configure>', lambda e: my_canvas.configure(scrollregion =
my_canvas.bbox("all")))

    second_frame = Frame(my_canvas)

    my_canvas.create_window((0,0), window=second_frame, anchor="nw")

    background=Canvas(second_frame,width=600,height=1445)

```

```
image=ImageTk.PhotoImage(Image.open("Specialists Page.jpg"))
```

```
label1=Label(image=image)
```

```
label1.image=image
```

```
background.create_image(0,0,anchor='nw',image=image)
```

```
background.pack(expand=True,fill=BOTH)
```

```
SDPage.geometry('600x1000')
```

```
def item_selected(event):
```

```
    selected_index = userOptions.curselection()
```

```
    for i in selected_index:
```

```
        if userOptions.get(i)=='Unsubscribe':
```

```
            unsubscribe()
```

```
        if userOptions.get(i)=='Logout':
```

```
            logout()
```

```
        if userOptions.get(i)=='T & C':
```

```
            TandC()
```

```
def button_hover1(a):
```

```
    diyaMoreButton["borderwidth"]=1
```

```
def button_hover_leave1(b):
```

```
    diyaMoreButton["borderwidth"]=0
```

```
def button_hover2(a):
```

```
tarunMatthewButton["borderwidth"]=1
```

```
def button_hover_leave2(b):
```

```
tarunMatthewButton["borderwidth"]=0
```

```
def button_hover3(a):
```

```
karimLokButton["borderwidth"]=1
```

```
def button_hover_leave3(b):
```

```
karimLokButton["borderwidth"]=0
```

```
def button_hover4(a):
```

```
vijayGaneshanButton["borderwidth"]=1
```

```
def button_hover_leave4(b):
```

```
vijayGaneshanButton["borderwidth"]=0
```

```
def button_hover5(a):
```

```
ananyaBahriButton["borderwidth"]=1
```

```
def button_hover_leave5(b):
```

```
ananyaBahriButton["borderwidth"]=0
```

```
def button_hover6(a):
```

```
radhikaRamanathanButton["borderwidth"]=1
```

```
def button_hover_leave6(b):
```

```
radhikaRamanathanButton["borderwidth"]=0
```

```
def button_hover7(a):
```

```
global userOptions

options= ('Unsubscribe','T & C','Logout')

options_var = tk.StringVar(value=options)

userOptions= tk.Listbox(second_frame,listvariable=options_var,height=3,selectmode='browse')

userOptions.place(x=450,y=50)

userOptions.bind('<<ListboxSelect>>',item_selected)

def button_hover_leave7(b):

    def button_hover8(a):

        userOptions['borderwidth']=1

    def button_hover_leave8(b):

        userOptions.destroy()

    userOptions.bind("<Enter>",button_hover8)

    userOptions.bind("<Leave>",button_hover_leave8)

back=ImageTk.PhotoImage(Image.open("Back-Button-Logo.jpg"))

backButtonLabel=Label(image=back)

backButtonLabel.image=back


backButton=Button(second_frame,image=back,borderwidth=0,highlightthickness=0,command=servic
es)

backButton.place(x=450,y=20)
```

```
user=ImageTk.PhotoImage(Image.open("User Logo.jpg"))
```

```
userButtonLabel=Label(image=user)
```

```
userButtonLabel.image=user
```

```
userButton=Button(second_frame,image=user,borderwidth=0,highlightthickness=0)
```

```
userButton.place(x=540,y=12)
```

```
diyaMore=ImageTk.PhotoImage(Image.open("Diya More.jpg"))
```

```
diyaMoreButtonLabel=Label(image=diyaMore)
```

```
diyaMoreButtonLabel.image=diyaMore
```

```
diyaMoreButton=Button(second_frame,image=diyaMore,command=specialConfirmation,borderwidth=0,highlightthickness=0)
```

```
diyaMoreButton.place(x=0,y=138)
```

```
tarunMatthew=ImageTk.PhotoImage(Image.open("Dr. Tarun Matthew.jpg"))
```

```
tarunMatthewButtonLabel=Label(image=tarunMatthew)
```

```
tarunMatthewButtonLabel.image=tarunMatthew
```

```
tarunMatthewButton=Button(second_frame,image=tarunMatthew,command=specialConfirmation,borderwidth=0,highlightthickness=0)
```

```
tarunMatthewButton.place(x=0,y=323)
```

```
karimLok=ImageTk.PhotoImage(Image.open("Karim Lok.jpg"))
```

```
karimLokButtonLabel=Label(image=karimLok)
```

```
karimLokButtonLabel.image=karimLok
```

```
karimLokButton=Button(second_frame,image=karimLok,command=specialConfirmation,borderwidth=0,highlightthickness=0)
```

```
karimLokButton.place(x=0,y=510)
```

```
vijayGaneshan=ImageTk.PhotoImage(Image.open("Vijay Ganeshan.jpg"))
```

```
vijayGaneshanButtonLabel=Label(image=vijayGaneshan)
```

```
vijayGaneshanButtonLabel.image=vijayGaneshan
```

```
vijayGaneshanButton=Button(second_frame,image=vijayGaneshan,command=specialConfirmation,borderwidth=0,highlightthickness=0)
```

```
vijayGaneshanButton.place(x=0,y=696)
```

```
ananyaBahri=ImageTk.PhotoImage(Image.open("Ananya Bahri.jpg"))
```

```
ananyaBahriButtonLabel=Label(image=ananyaBahri)
```

```
ananyaBahriButtonLabel.image=ananyaBahri
```

```
ananyaBahriButton=Button(second_frame,image=ananyaBahri,command=specialConfirmation,borderwidth=0,highlightthickness=0)
```

```
ananyaBahriButton.place(x=0,y=879)
```

```
radhikaRamanathan=ImageTk.PhotoImage(Image.open("Radhika Ramanathan.jpg"))
```

```
radhikaRamanathanButtonLabel=Label(image=radhikaRamanathan)
```

```
radhikaRamanathanButtonLabel.image=radhikaRamanathan
```

```
radhikaRamanathanButton=Button(second_frame,image=radhikaRamanathan,command=specialConfi  
rmation,borderwidth=0,highlightthickness=0)
```

```
radhikaRamanathanButton.place(x=0,y=1064)
```

```
diyaMoreButton.bind("<Enter>",button_hover1)
```

```
diyaMoreButton.bind("<Leave>",button_hover_leave1)
```

```
tarunMatthewButton.bind("<Enter>",button_hover2)
```

```
tarunMatthewButton.bind("<Leave>",button_hover_leave2)
```

```
karimLokButton.bind("<Enter>",button_hover3)
```

```
karimLokButton.bind("<Leave>",button_hover_leave3)
```

```
vijayGaneshanButton.bind("<Enter>",button_hover4)
```

```
vijayGaneshanButton.bind("<Leave>",button_hover_leave4)
```

```
ananyaBahriButton.bind("<Enter>",button_hover5)
```

```
ananyaBahriButton.bind("<Leave>",button_hover_leave5)
```

```
radhikaRamanathanButton.bind("<Enter>",button_hover6)
```

```
radhikaRamanathanButton.bind("<Leave>",button_hover_leave6)
```

```
userButton.bind("<Enter>",button_hover7)
```



```
userButton.bind("<Leave>",button_hover_leave7)
```

```
#General Doctors Confirmation Page
```

```
def generalDoctorsConfirmation():
```

```
    def generalDoctorsInvoice(recievedValue):
```

```
        Name=fullName.title()
```

```
        a=recievedValue.split(", ₹")
```

```
        cost=int(a[1])
```

```
        def grab_date():
```

```
            date=cal.get_date()
```

```
            return date
```

```
        date=grab_date()
```

```
        listForDate=date.split('/')
```

```
        finalDate=listForDate[1]+'-'+listForDate[0]+'-'+listForDate[2]
```

```
        price="₹"+str((cost*(2/100))+cost+((10/100)*cost))
```

```
        GDIPage=Toplevel(MPage)
```

```
        background=Canvas(GDIPage,width=600,height=600)
```

```
        image=ImageTk.PhotoImage(Image.open("Invoice Page.jpg"))
```

```
        label1=Label(image=image)
```

```
        label1.image=image
```

```

background.create_image(0,0,anchor='nw',image=image)

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

GDIPage.geometry('600x600')

nameLabel=Label(GDIPage, text=Name,width=21,font=('Bahnschrift
Condensed',20),bg='black',fg='gold')

nameLabel.place(x=305,y=180)

specialLabel=Label(GDIPage,text=recievedValue,width=21,font=('Bahnschrift
Condensed',20),bg='black',fg='gold')

specialLabel.place(x=305,y=262)

priceLabel=Label(GDIPage,text=price,width=21,font=('Bahnschrift
Condensed',20),bg='black',fg='gold')

priceLabel.place(x=305,y=342)

dateLabel=Label(GDIPage,text=finalDate,font=('Bahnschrift
Condensed',22),bg='black',fg='gold')

dateLabel.place(x=375,y=424)

done=Button(GDIPage, text="Done",font=('Bahnschrift
Condensed',18),bg='black',fg='gold',command=logout)

done.place(x=265, y=480)

invoiceNumber=Label(GDIPage,text='Invoice No. :
'+str(random.randint(0,10000000)),font=('Bahnschrift Condensed',16),bg='black',fg='gold')

```

```
invoiceNumber.place(x=420, y=20)

def selected():

    a=var.get()

    submit=ImageTk.PhotoImage(Image.open("Submit Button.jpg"))

    submitButtonLabel=Label(image=submit)

    submitButtonLabel.image=submit


submitButton=Button(GDCPage,image=submit,borderwidth=0,highlightthickness=0,command=partial
(generalDoctorsInvoice,a))

    submitButton.place(x=230,y=620)

GDCPage=Toplevel(MPage)

background=Canvas(GDCPage,width=600,height=700)

image=ImageTk.PhotoImage(Image.open("Confirmation Page.jpg"))

label1=Label(image=image)

label1.image=image

background.create_image(0,0,anchor='nw',image=image)

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

GDCPage.geometry('600x700')

cal=Calendar(GDCPage,selectmode="day",year=2022,month=8,day=22)

cal.place(x=25,y=310)
```

```

var=tk.StringVar()

um=Radiobutton(GDCPage,text='Usha Manda-Rs2099/session',font=('Bahnschrift
Condensed',16),variable=var,value='Usha Manda,
₹2199',command=selected,tristatevalue=0,bg="black",fg="gold")

um.place(x=315,y=280)

jv=Radiobutton(GDCPage,text='John Varghese-Rs2199/session',font=('Bahnschrift
Condensed',16),variable=var,value='John Varghese,
₹2499',command=selected,tristatevalue=0,bg="black",fg="gold")

jv.place(x=315,y=330)

jr=Radiobutton(GDCPage,text='Jasmine Raju-Rs2299/session',font=('Bahnschrift
Condensed',16),variable=var,value='Jasmine Raju,
₹2299',command=selected,tristatevalue=0,bg="black",fg="gold")

jr.place(x=315,y=380)

ar=Radiobutton(GDCPage,text='Anand Radhakrishnan-Rs2399/session',font=('Bahnschrift
Condensed',16),variable=var,value='Anand R,
₹2399',command=selected,tristatevalue=0,bg="black",fg="gold")

ar.place(x=302,y=430)

#General Doctors Page

def generaldoctors():

```

```
GDPPage=Toplevel(MPage)

main_frame=Frame(GDPPage)

main_frame.pack(fill=BOTH, expand=1)

my_canvas = Canvas(main_frame)

my_canvas.pack(side=LEFT, fill=BOTH, expand=1)

my_scrollbar = ttk.Scrollbar(main_frame, orient=VERTICAL, command=my_canvas.yview)

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

my_canvas.configure(yscrollcommand=my_scrollbar.set)

my_canvas.bind('<Configure>', lambda e: my_canvas.configure(scrollregion =

my_canvas.bbox("all")))

second_frame = Frame(my_canvas)

my_canvas.create_window((0,0), window=second_frame, anchor="nw")

background=Canvas(second_frame,width=600,height=1045)

image=ImageTk.PhotoImage(Image.open("General Doctors Page.jpg"))

label1=Label(image=image)

label1.image=image

background.create_image(0,0,anchor='nw',image=image)

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

GDPPage.geometry('600x1000')

def item_selected(event):
```

```
selected_index = userOptions.curselection()

for i in selected_index:

    if userOptions.get(i)=='Unsubscribe':

        unsubscribe()

    if userOptions.get(i)=='Logout':

        logout()

    if userOptions.get(i)=='T & C':

        TandC()

def button_hover1(a):

    johnVargheseButton["borderwidth"]=1

def button_hover_leave1(b):

    johnVargheseButton["borderwidth"]=0

def button_hover2(a):

    ushaMandaButton["borderwidth"]=1

def button_hover_leave2(b):

    ushaMandaButton["borderwidth"]=0

def button_hover3(a):

    jasminRajuButton["borderwidth"]=1

def button_hover_leave3(b):

    jasminRajuButton["borderwidth"]=0
```

```
def button_hover4(a):

    anandRadhakishanButton["borderwidth"]=1

def button_hover_leave4(b):

    anandRadhakishanButton["borderwidth"]=0

def button_hover7(a):

    global userOptions

    options= ('Unsubscribe','T & C','Logout')

    options_var = tk.StringVar(value=options)

    userOptions= tk.Listbox(second_frame,listvariable=options_var,height=3,selectmode='browse')

    userOptions.place(x=450,y=50)

    userOptions.bind('<<ListboxSelect>>',item_selected)

def button_hover_leave7(b):

    def button_hover8(a):

        userOptions['borderwidth']=1

    def button_hover_leave8(b):

        userOptions.destroy()

    userOptions.bind("<Enter>",button_hover8)

    userOptions.bind("<Leave>",button_hover_leave8)

back=ImageTk.PhotoImage(Image.open("Back-Button-Logo.jpg"))

backButtonLabel=Label(image=back)
```

```
backButtonLabel.image=back
```

```
backButton=Button(second_frame,image=back,borderwidth=0,highlightthickness=0,command=servic  
es)
```

```
backButton.place(x=450,y=20)
```

```
user=ImageTk.PhotoImage(Image.open("User Logo.jpg"))
```

```
userButtonLabel=Label(image=user)
```

```
userButtonLabel.image=user
```

```
userButton=Button(second_frame,image=user,borderwidth=0,highlightthickness=0)
```

```
userButton.place(x=540,y=12)
```

```
johnVarghese=ImageTk.PhotoImage(Image.open("John Varghese.jpg"))
```

```
johnVargheseButtonLabel=Label(image=johnVarghese)
```

```
johnVargheseButtonLabel.image=johnVarghese
```

```
johnVargheseButton=Button(second_frame,image=johnVarghese,borderwidth=0,highlightthickness=0  
,command=generalDoctorsConfirmation)
```

```
johnVargheseButton.place(x=0,y=138)
```

```
ushaManda=ImageTk.PhotoImage(Image.open("Usha Manda.jpg"))
```

```
ushaMandaButtonLabel=Label(image=ushaManda)
```

```
ushaMandaButtonLabel.image=ushaManda
```



```
ushaMandaButton=Button(second_frame,image=ushaManda,borderwidth=0,highlightthickness=0,command=generalDoctorsConfirmation)

ushaMandaButton.place(x=0,y=323)

jasminRaju=ImageTk.PhotoImage(Image.open("Jasmin Raju.jpg"))

jasminRajuButtonLabel=Label(image=jasminRaju)

jasminRajuButtonLabel.image=jasminRaju

jasminRajuButton=Button(second_frame,image=jasminRaju,borderwidth=0,highlightthickness=0,command=generalDoctorsConfirmation)

jasminRajuButton.place(x=0,y=510)

anandRadhakishan=ImageTk.PhotoImage(Image.open("Anand Radhakishan.jpg"))

anandRadhakishanButtonLabel=Label(image=anandRadhakishan)

anandRadhakishanButtonLabel.image=anandRadhakishan

anandRadhakishanButton=Button(second_frame,image=anandRadhakishan,borderwidth=0,highlightthickness=0,command=generalDoctorsConfirmation)

anandRadhakishanButton.place(x=0,y=696)

johnVargheseButton.bind("<Enter>",button_hover1)

johnVargheseButton.bind("<Leave>",button_hover_leave1)
```

```
ushaMandaButton.bind("<Enter>",button_hover2)

ushaMandaButton.bind("<Leave>",button_hover_leave2)

jasminRajuButton.bind("<Enter>",button_hover3)

jasminRajuButton.bind("<Leave>",button_hover_leave3)

anandRadhakishanButton.bind("<Enter>",button_hover4)

anandRadhakishanButton.bind("<Leave>",button_hover_leave4)

userButton.bind("<Enter>",button_hover7)

userButton.bind("<Leave>",button_hover_leave7)
```

#Available Services Page

```
def services():
```

```
    SPage=Toplevel(MPage)

    background=Canvas(SPage,width=600,height=600)

    image=ImageTk.PhotoImage(Image.open("Services Page.jpg"))

    label1=Label(image=image)

    label1.image=image

    background.create_image(0,0,anchor='nw',image=image)

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

    SPage.geometry('600x600')

    def item_selected(event):
```

```
selected_index = userOptions.curselection()

for i in selected_index:

    if userOptions.get(i)=='Unsubscribe':

        unsubscribe()

    if userOptions.get(i)=='Logout':

        logout()

    if userOptions.get(i)=='T & C':

        TandC()

def button_hover1(a):

    generalDoctorsButton["borderwidth"]=1

def button_hover_leave1(b):

    generalDoctorsButton["borderwidth"]=0

def button_hover2(a):

    specialDoctorsButton["borderwidth"]=1

def button_hover_leave2(b):

    specialDoctorsButton["borderwidth"]=0

def button_hover3(a):

    workshopsButton["borderwidth"]=1

def button_hover_leave3(b):

    workshopsButton["borderwidth"]=0
```

```
def button_hover7(a):

    global userOptions

    options= ('Unsubscribe','T & C','Logout')

    options_var = tk.StringVar(value=options)

    userOptions= tk.Listbox(SPage,listvariable=options_var,height=3,selectmode='browse')

    userOptions.place(x=450,y=50)

    userOptions.bind('<<ListboxSelect>>',item_selected)

def button_hover_leave7(b):

    def button_hover8(a):

        userOptions['borderwidth']=1

    def button_hover_leave8(b):

        userOptions.destroy()

    userOptions.bind("<Enter>",button_hover8)

    userOptions.bind("<Leave>",button_hover_leave8)

back=ImageTk.PhotoImage(Image.open("Back-Button-Logo.jpg"))

backButtonLabel=Label(image=back)

backButton=Button(SPage,image=back,borderwidth=0,highlightthickness=0,command=welcome)

backButton.place(x=450,y=20)

backButtonLabel.image=back

user=ImageTk.PhotoImage(Image.open("User Logo.jpg"))
```

```
userButtonLabel=Label(image=user)
```

```
userButtonLabel.image=user
```

```
userButton=Button(SPage,image=user,borderwidth=0,highlightthickness=0)
```

```
userButton.place(x=550,y=12)
```

```
generalDoctors=ImageTk.PhotoImage(Image.open("General Doctors Label.jpg"))
```

```
generalDoctorsButtonLabel=Label(image=generalDoctors)
```

```
generalDoctorsButtonLabel.image=generalDoctors
```

```
generalDoctorsButton=Button(SPage,image=generalDoctors,command=generaldoctors,borderwidth=0,highlightthickness=0)
```

```
generalDoctorsButton.place(x=0,y=170)
```

```
specialDoctors=ImageTk.PhotoImage(Image.open("Specialists Label.jpg"))
```

```
specialDoctorsButtonLabel=Label(image=specialDoctors)
```

```
specialDoctorsButtonLabel.image=specialDoctors
```

```
specialDoctorsButton=Button(SPage,image=specialDoctors,command=specialdoctors,borderwidth=0,highlightthickness=0)
```

```
specialDoctorsButton.place(x=0,y=309)
```

```
workshops=ImageTk.PhotoImage(Image.open("Workshops Label.jpg"))
```

```
workshopsButtonLabel=Label(image=workshops)
```

```
workshopsButtonLabel.image=workshops
```

```
workshopsButton=Button(SPage,image=workshops,command=workshop,borderwidth=0,highlightthickness=0)
```

```
workshopsButton.place(x=0,y=447)
```

```
generalDoctorsButton.bind("<Enter>",button_hover1)
```

```
generalDoctorsButton.bind("<Leave>",button_hover_leave1)
```

```
specialDoctorsButton.bind("<Enter>",button_hover2)
```

```
specialDoctorsButton.bind("<Leave>",button_hover_leave2)
```

```
workshopsButton.bind("<Enter>",button_hover3)
```

```
workshopsButton.bind("<Leave>",button_hover_leave3)
```

```
userButton.bind("<Enter>",button_hover7)
```

```
userButton.bind("<Leave>",button_hover_leave7)
```

```
#About Us Page
```

```
def aboutUs():
```

```
    AUPage=Toplevel(MPage)
```

```
    background=Canvas(AUPage,width=600,height=600)
```

```
    image=ImageTk.PhotoImage(Image.open("About Us Page.jpg"))
```

```
    label1=Label(image=image)
```

```
label1.image=image

background.create_image(0,0,anchor='nw',image=image)

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

AUPage.geometry('600x600')

back=ImageTk.PhotoImage(Image.open("Back-Button-Logo.jpg"))

backButtonLabel=Label(image=back)

backButtonLabel.image=back

backButton=Button(AUPage,image=back,borderwidth=0,highlightthickness=0,command=welcome)

backButton.place(x=450,y=15)

def item_selected(event):

    selected_index = userOptions.curselection()

    for i in selected_index:

        if userOptions.get(i)=='Unsubscribe':

            unsubscribe()

        if userOptions.get(i)=='Logout':

            logout()

        if userOptions.get(i)=='T & C':

            TandC()

def button_hover7(a):

    global userOptions
```

```
options= ('Unsubscribe','T & C','Logout')

options_var = tk.StringVar(value=options)

userOptions= tk.Listbox(AUPage,listvariable=options_var,height=3,selectmode='browse')

userOptions.place(x=450,y=50)

userOptions.bind('<<ListboxSelect>>',item_selected)

def button_hover_leave7(b):

    def button_hover8(a):

        userOptions['borderwidth']=1

    def button_hover_leave8(b):

        userOptions.destroy()

    userOptions.bind("<Enter>",button_hover8)

    userOptions.bind("<Leave>",button_hover_leave8)

user=ImageTk.PhotoImage(Image.open("User Logo.jpg"))

userButtonLabel=Label(image=user)

userButtonLabel.image=user

userButton=Button(AUPage,image=user,borderwidth=0,highlightthickness=0)

userButton.place(x=550,y=8)

userButton.bind("<Enter>",button_hover7)

userButton.bind("<Leave>",button_hover_leave7)
```



```
#Welcome Page
```

```
def welcome():
```

```
    WPage=Toplevel(MPage)
```

```
    background=Canvas(WPage,width=600,height=600)
```

```
    image=ImageTk.PhotoImage(Image.open("Welcome Page.jpg"))
```

```
    label1=Label(image=image)
```

```
    label1.image=image
```

```
    background.create_image(0,0,anchor='nw',image=image)
```

```
    background.pack(expand=True,fill=BOTH)
```

```
    WPage.geometry('600x600')
```

```
def item_selected(event):
```

```
    selected_index = userOptions.curselection()
```

```
    for i in selected_index:
```

```
        if userOptions.get(i)=='Unsubscribe':
```

```
            unsubscribe()
```

```
        if userOptions.get(i)=='Logout':
```

```
            logout()
```

```
        if userOptions.get(i)=='T & C':
```

```
            TandC()
```

```
def button_hover7(a):
```

```
global userOptions

options= ('Unsubscribe','T & C','Logout')

options_var = tk.StringVar(value=options)

userOptions= tk.Listbox(WPage,listvariable=options_var,height=3,selectmode='browse')

userOptions.place(x=450,y=50)

userOptions.bind('<<ListboxSelect>>',item_selected)

def button_hover_leave7(b):

    def button_hover8(a):

        userOptions['borderwidth']=1

    def button_hover_leave8(b):

        userOptions.destroy()

    userOptions.bind("<Enter>",button_hover8)

    userOptions.bind("<Leave>",button_hover_leave8)

user=ImageTk.PhotoImage(Image.open("User Logo.jpg"))

userButtonLabel=Label(image=user)

userButtonLabel.image=user

userButton=Button(WPage,image=user,borderwidth=0,highlightthickness=0)

userButton.place(x=550,y=8)

continueButton=Button(WPage, text="CONTINUE",font=('Bahnschrift
Condensed',24),bg='black',fg='gold',borderwidth=0,command=services)
```

```
continueButton.place(x=240,y=400)
```

```
aboutUsButton=Button(WPage,text='ABOUT US',font=('Bahnschrift  
Condensed',15),bg='black',fg='gold',borderwidth=0,command=aboutUs)
```

```
aboutUsButton.place(x=50,y=530)
```

```
userButton.bind("<Enter>",button_hover7)
```

```
userButton.bind("<Leave>",button_hover_leave7)
```

```
#Register Page
```

```
def register():
```

```
    def enter():
```

```
        usid=0
```

```
        psid=0
```

```
        if firstname.get()==" or lastname.get()==" or email.get()==" or username.get()==" or  
password.get()==":
```

```
            messagebox.showerror("Error","All Fields are Required")
```

```
connection = mysql.connector.connect(
```

```
    database='mindcology',
```

```
    user='root',
```

```
    password='12ammu34')
```

```
sql_select_Query = "select * from userinfo"
```

```

cursor = connection.cursor()

cursor.execute(sql_select_Query)

records = cursor.fetchall()

for row in records:

    if row[2] == username.get():

        messagebox.showinfo('Error','Username Already in Use')

        usid+=1

    if row[3]==password.get():

        messagebox.showinfo('Error','Password Already in Use')

        psid+=1

if usid==0:

    if psid==0:

        mydb = mysql.connector.connect(host="localhost", user="root", password="12ammu34",
database="mindcology")

        mycursor = mydb.cursor()

        sql = "INSERT INTO userinfo (FIRSTNAME, LASTNAME, USERNAME, PASSWORD) VALUES
(%s, %s,%s,%s)"

        val = (firstname.get(), lastname.get(),username.get(),password.get())

        mycursor.execute(sql, val)

        sql2="INSERT INTO mailing_list (FIRSTNAME, LASTNAME, EMAIL) VALUES (%s, %s,%s)"

```

```
val2=(firstname.get(),lastname.get(),email.get())

mycursor.execute(sql2, val2)

mydb.commit()

messagebox.showinfo("Congrats","Registration Successfull")

welcome()

nameForInvoice(firstname.get(),lastname.get())

large_font=('Bahnschrift Condensed',20)

RPage=Toplevel(MPage)

background=Canvas(RPage,width=600,height=600)

image=ImageTk.PhotoImage(Image.open("Registration Page.jpg"))

label1=Label(image=image)

label1.image=image

background.create_image(0,0,anchor='nw',image=image)

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

RPage.geometry('600x600')

def item_selected(event):

    selected_index = userOptions.curselection()

    for i in selected_index:

        if userOptions.get(i)=='T & C':

            TandC()
```

```
def button_hover7(a):

    global userOptions

    options= ('T & C')

    options_var = tk.StringVar(value=options)

    userOptions= tk.Listbox(RPage,listvariable=options_var,height=3,selectmode='browse')

    userOptions.place(x=450,y=50)

    userOptions.bind('<<ListboxSelect>>',item_selected)

def button_hover_leave7(b):

    def button_hover8(a):

        userOptions['borderwidth']=1

    def button_hover_leave8(b):

        userOptions.destroy()

    userOptions.bind("<Enter>",button_hover8)

    userOptions.bind("<Leave>",button_hover_leave8)

user=ImageTk.PhotoImage(Image.open("User Logo.jpg"))

userButtonLabel=Label(image=user)

userButtonLabel.image=user

userButton=Button(RPage,image=user,command=user,borderwidth=0,highlightthickness=0)

userButton.place(x=550,y=8)
```

```
firstname=Entry(RPage,font=large_font,bg="black",fg="white",width=24,insertbackground="white")
```

```
    firstname.place(x=250,y=156)
```

```
    firstname.bind("<Return>",enter)
```

```
lastname=Entry(RPage,font=large_font,bg="black",fg="white",width=24,insertbackground="white")
```

```
    lastname.place(x=250,y=228)
```

```
    lastname.bind("<Return>",enter)
```

```
email=Entry(RPage,font=large_font,bg="black",fg="white",width=24,insertbackground="white")
```

```
    email.place(x=250,y=298)
```

```
    email.bind("<Return>",enter)
```

```
username=Entry(RPage,font=large_font,bg="black",fg="white",width=24,insertbackground="white")
```

```
    username.place(x=250,y=370)
```

```
    username.bind("<Return>",enter)
```

```
password=Entry(RPage,font=large_font,bg="black",fg="white",width=24,insertbackground="white")
```

```
    password.place(x=250,y=439)
```

```
    password.bind("<Return>",enter)
```

```
submit=ImageTk.PhotoImage(Image.open("Submit Button.jpg"))
```

```
submitButtonLabel=Label(image=submit)
```

```
submitButtonLabel.image=submit
```

```
submitButton=Button(RPage,image=submit,borderwidth=0,highlightthickness=0,command=enter)
```

```
submitButton.place(x=235,y=511)
```

```
userButton.bind("<Enter>",button_hover7)
```

```
userButton.bind("<Leave>",button_hover_leave7)
```

#Login Page

```
def login():
```

```
    def entry():
```

```
        error=0
```

```
        firstnameSend=""
```

```
        lastnameSend=""
```

```
        connection = mysql.connector.connect(
```

```
            database='mindcology',
```

```
            user='root',
```

```
            password='12ammu34')
```

```
        sql_select_Query = "select * from userinfo"
```

```
        cursor = connection.cursor()
```

```
        cursor.execute(sql_select_Query)
```



```
records = cursor.fetchall()

for row in records:

    if row[2] != username.get() or row[3]!=password.get():

        continue

    else:

        firstnameSend=row[0]

        lastnameSend=row[1]

        break

if firstnameSend==" or lastnameSend=="':

    messagebox.showinfo('Error','Username or Password  Incorrect')

else:

    welcome()

    nameForInvoice(firstnameSend,lastnameSend)

large_font=('Bahnschrift Condensed',20)

LPage=Toplevel(MPage)

background=Canvas(LPage,width=600,height=600)

image=ImageTk.PhotoImage(Image.open("Login Page.jpg"))

label1=Label(image=image)

label1.image=image

background.create_image(0,0,anchor='nw',image=image)
```

```
background.pack(expand=True,fill=BOTH)

LPage.geometry('600x600')

def item_selected(event):

    selected_index = userOptions.curselection()

    for i in selected_index:

        if userOptions.get(i)=='Unsubscribe':

            unsubscribe()

        if userOptions.get(i)=='Logout':

            logout()

        if userOptions.get(i)=='T & C':

            TandC()

def button_hover7(a):

    global userOptions

    options= ('Unsubscribe','T & C','Logout')

    options_var = tk.StringVar(value=options)

    userOptions=

tk.Listbox(LPage,listvariable=options_var,height=3,selectmode='browse',relief='solid')

    userOptions.place(x=450,y=50)

    userOptions.bind('<<ListboxSelect>>',item_selected)

def button_hover_leave7(b):
```

```
def button_hover8(a):

    userOptions['borderwidth']=1

def button_hover_leave8(b):

    userOptions.destroy()

    userOptions.bind("<Enter>",button_hover8)

    userOptions.bind("<Leave>",button_hover_leave8)

user=ImageTk.PhotoImage(Image.open("User Logo.jpg"))

userButtonLabel=Label(image=user)

userButtonLabel.image=user

userButton=Button(LPage,image=user,command=user,borderwidth=0,highlightthickness=0)

userButton.place(x=550,y=8)


username=Entry(LPage,font=large_font,bg="black",fg="white",width=24,insertbackground="white")

username.place(x=250,y=233)

username.bind("<Return>",entry)


password=Entry(LPage,font=large_font,bg="black",fg="white",width=24,insertbackground="white",show="*")

password.place(x=250,y=344)

password.bind("Return",entry)
```

```
submit=ImageTk.PhotoImage(Image.open("Submit Button.jpg"))

submitButtonLabel=Label(image=submit)

submitButtonLabel.image=submit

submitButton=Button(LPage,image=submit,borderwidth=0,highlightthickness=0,command=entry)

submitButton.place(x=235,y=470)

userButton.bind("<Enter>",button_hover7)

userButton.bind("<Leave>",button_hover_leave7)
```

#Main Page

```
MPage=tk.Tk()

background=Canvas(MPage,width=600,height=338)

image=ImageTk.PhotoImage(Image.open("Main-Page-Background.jpg"))

background.create_image(0,0,anchor='nw',image=image)

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

MPage.title('Mindcology')

MPage.geometry('600x600')

MPage.configure(bg='white')

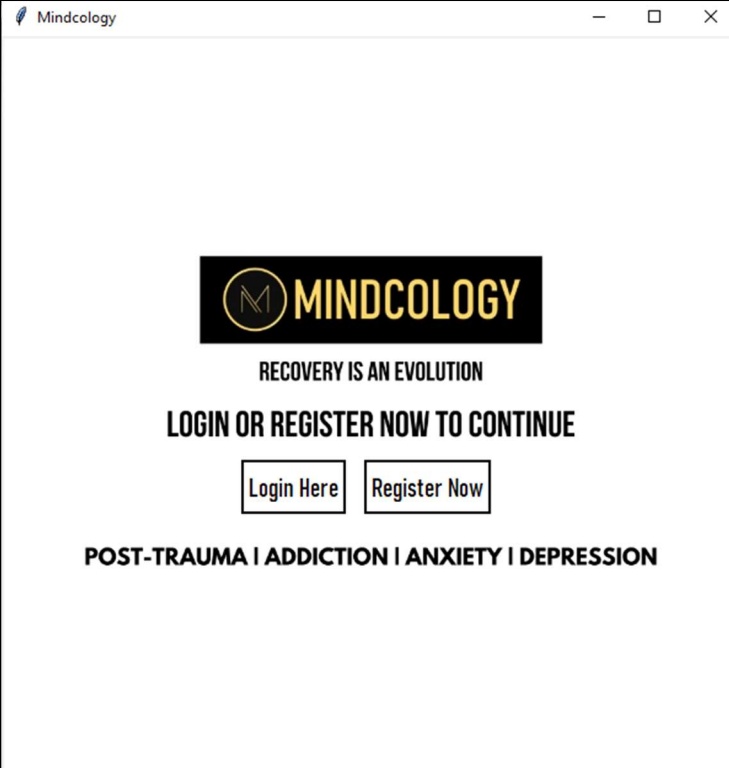
Login=Button(MPage,text='Login Here',font=('Bahnschrift
Condensed',16),bg='white',fg='black',borderwidth=2, relief="solid",command=login)

Login.place(x=195, y=345)
```

```
Register=Button(MPage,text='Register Now',font=('Bahnschrift  
Condensed',16),bg='white',fg='black',borderwidth=2, relief="solid",command=register)  
  
Register.place(x=295, y=345)  
  
MPage.mainloop()
```

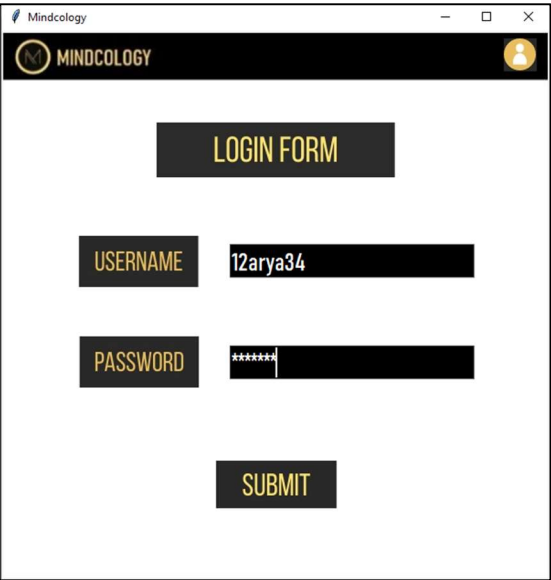
OUTPUT AND SCREENSHOTS

Main Page

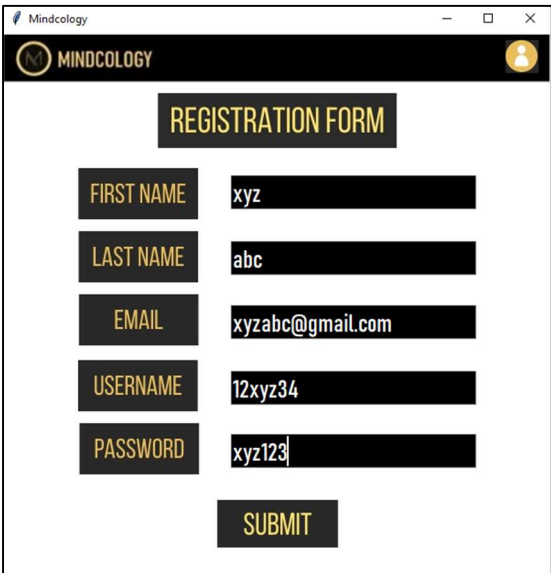


A screenshot of a web browser window titled "Mindcology". The page features a central logo with a stylized "M" inside a circle, followed by the word "MINDCOLOGY" in bold, yellow, uppercase letters. Below the logo, the text "RECOVERY IS AN EVOLUTION" is displayed in a smaller, black, uppercase font. This is followed by the prompt "LOGIN OR REGISTER NOW TO CONTINUE" in bold, black, uppercase letters. Two buttons, "Login Here" and "Register Now", are positioned below the prompt. At the bottom of the page, the text "POST-TRAUMA | ADDICTION | ANXIETY | DEPRESSION" is shown in bold, black, uppercase letters.

Login and Register Pages

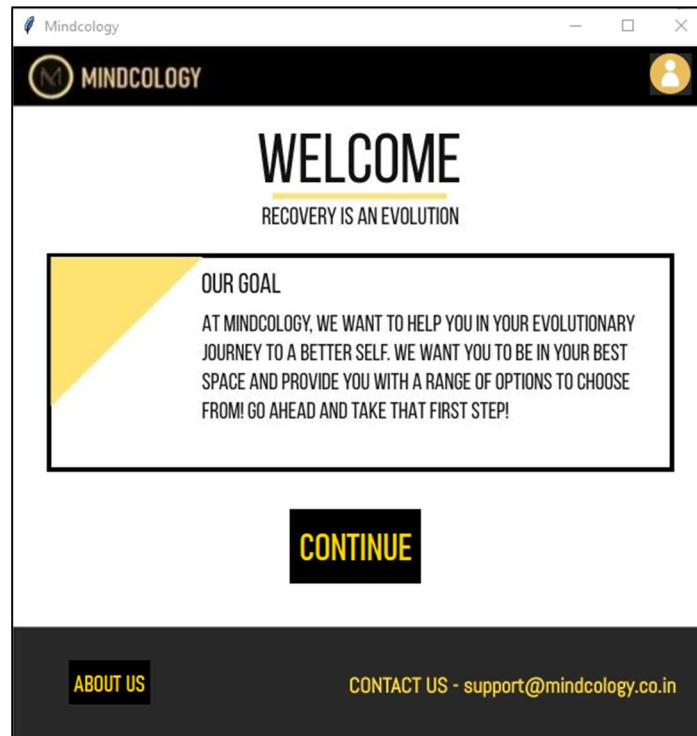


A screenshot of a web browser window titled "Mindcology" showing the login form. The header includes the Mindcology logo and a user profile icon. The form is titled "LOGIN FORM" in bold, yellow, uppercase letters. It contains two input fields: "USERNAME" with the value "12arya34" and "PASSWORD" with masked characters "*****". A "SUBMIT" button is located at the bottom of the form.

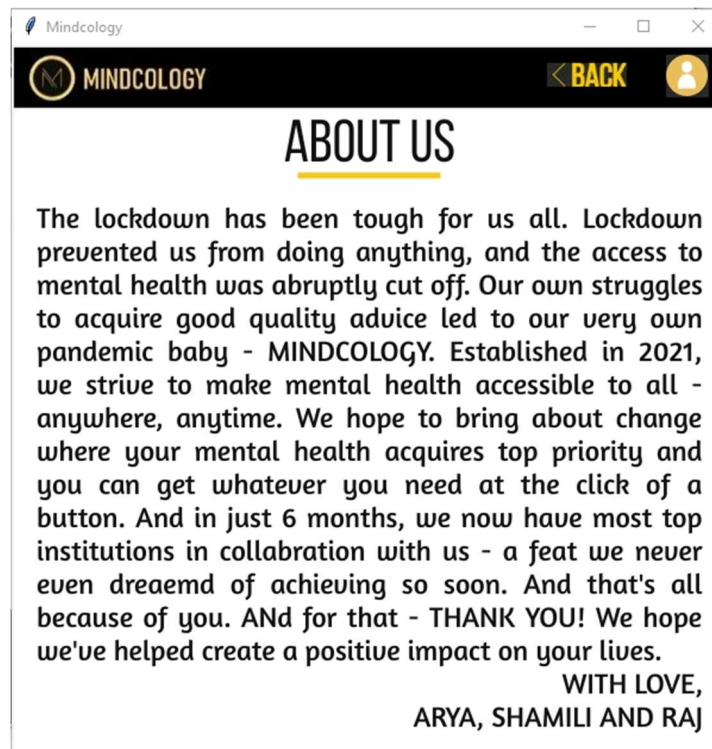


A screenshot of a web browser window titled "Mindcology" showing the registration form. The header includes the Mindcology logo and a user profile icon. The form is titled "REGISTRATION FORM" in bold, yellow, uppercase letters. It contains five input fields: "FIRST NAME" with the value "xyz", "LAST NAME" with the value "abc", "EMAIL" with the value "xyzabc@gmail.com", "USERNAME" with the value "12xyz34", and "PASSWORD" with the value "xyz123". A "SUBMIT" button is located at the bottom of the form.

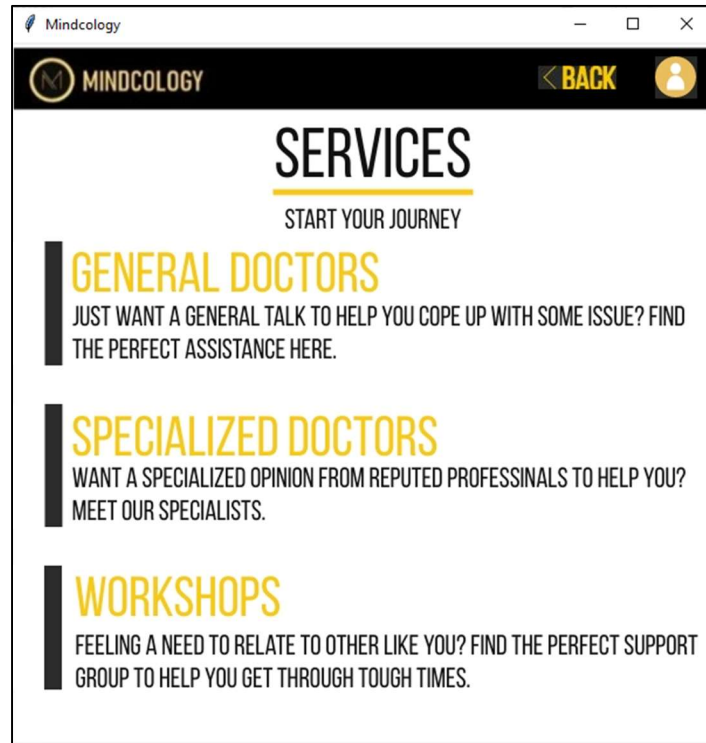
Welcome Page



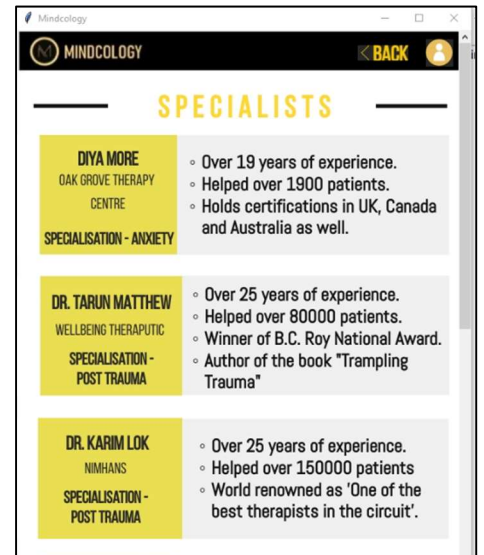
About Us Page



Available Services Page



General Doctors, Specialists and Workshops Pages



Booking Confirmation Page

Mindcology

 MINDCOLOGY

CONFIRMATION

CONFIRM YOUR BOOKING

CHOOSE YOUR DATE

January 2022

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
52	27	28	29	30	31	1	2
1	3	4	5	6	7	8	9
2	10	11	12	13	14	15	16
3	17	18	19	20	21	22	23
4	24	25	26	27	28	29	30
5	31	1	2	3	4	5	6


CONFIRM YOUR WORKSHOP

- Addiction Extinction - Rs1499/session
- Absolut Anxiety - Rs1499/session
- Against Trauma - Rs1499/session
- Dawn Over Depression- Rs1499/session
- Feeling and Healing - Rs1499/session
- Adios Addiction - Rs1499/session

SUBMIT

Invoice Page

Mindcology

 MINDCOLOGY

Invoice No. : 7003672

INVOICE

FULLNAME OF PATIENT

Arya Hariharan

WORKSHOP

Dawn Over Depression

PRICE

₹1678.88

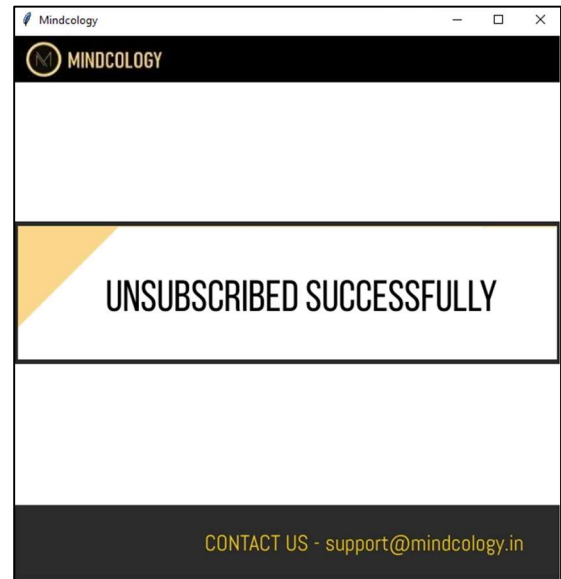
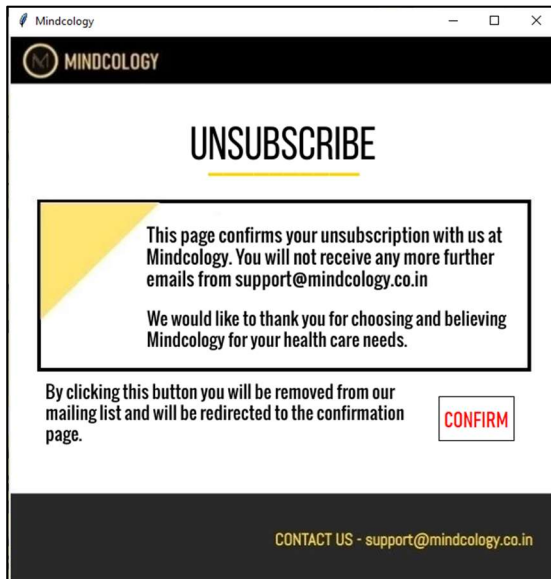
DATE

7-1-22

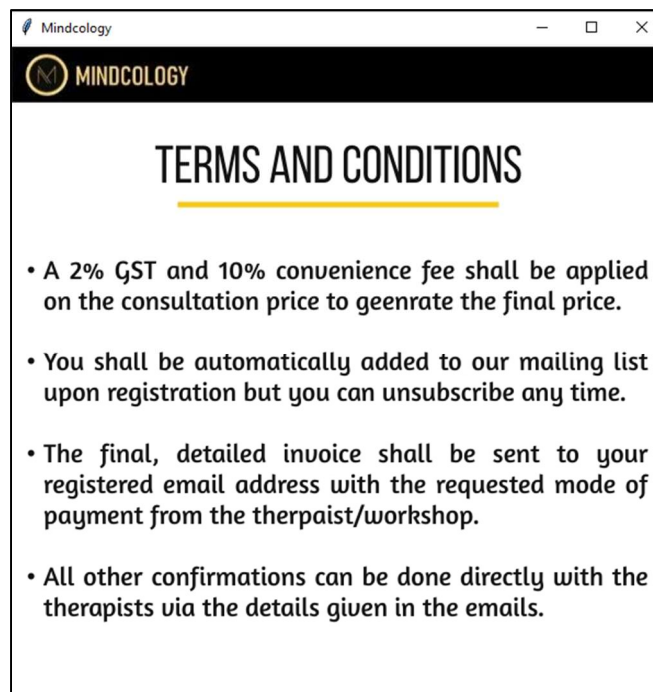
Done

*A CONFIRMATION EMAIL WITH MODE OF PAYMENT SHALL BE SENT DIRECTLY TO YOU FROM THE CHOSEN AGENCY.

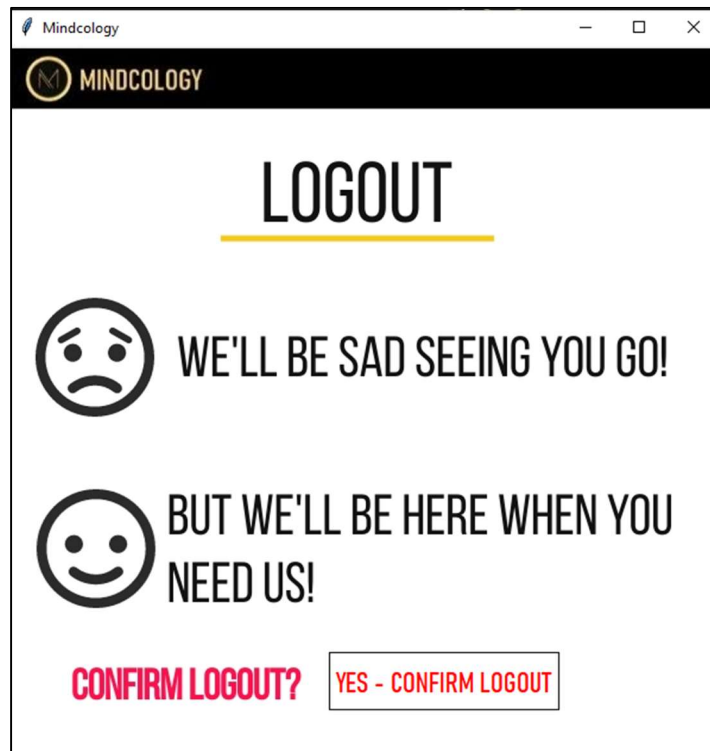
Unsubscribe Pages



Terms and Conditions Page



Logout Pages



SQL TABLES

```
mysql> select* from userinfo;
+-----+-----+-----+-----+
| FIRSTNAME | LASTNAME | USERNAME | PASSWORD |
+-----+-----+-----+-----+
| raj       | ammu     | 12raj34  | raj123   |
| arya      | hariharan | 12arya34 | arya123  |
| shamili   | gande    | 12shamili34 | shamili123 |
| sonia     | chacko   | 12sonia34 | sonia123 |
+-----+-----+-----+-----+
4 rows in set (0.01 sec)

mysql> select * from mailing_list;
+-----+-----+-----+
| FIRSTNAME | LASTNAME | EMAIL |
+-----+-----+-----+
| shamili   | gande    | shamiligande@gmail.com |
| sonia     | chacko   | sonia_maam@gmail.com   |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

CONCLUSIONS

This project focuses upon creating an application using the Python programming language. The project is titled Mindcology and is a booking system which one can use to book appointments with doctors, therapists and support groups. The project focuses on building accessibility to mental health resources. The project proved to be immensely insightful into the world of Python and its various applications.

LIMITATIONS

1. The program could not include an in-built payment method.
2. The scrollbar could not be associated with the scrolling motion on the mouse.
3. The hover features were primitive.

BIBLIOGRAPHY

“Stack Overflow.” *Stack Overflow*, stackoverflow.com

“Codemy.Com.” *Codemy*, codemy.com

<https://docs.python.org> › library › tkinter