

**COMPUTER SCIENCE**

**PROJECT**

*Bus Ticket Booking System*

Name: Sri Devi C.R.

Class: XII

Combination: PCMCs

**CERTIFICATE**

Name: Sri Devi C.R.

Class: XII

Combination: PCMCs

School: Alpine Public School

This is certified to be bona fide work of the student in the computer science laboratory during academic year 2020-2021.

Teacher in charge signature:

Examiner’s Signature:

Principal’s Signature:

School’s seal:

Date:

**ACKNOWLEGMENT**

I would like to express my deepest appreciation to all those who provided me with the possibility to complete this project. I would like to thank my computer science teacher, Mrs. Deepika N.S., whose valuable guidance has helped me complete this project. Her suggestions and instructions have served as a major contributor towards the completion of the project.

I take this opportunity to thank our head of the institution, Mrs. Jaya Shastry, who was always supportive and helpful in fulfilling all our academic requirements.

**CONTENTS**

* Introduction
* Program Design
* Source Code
* Bibliography
* Output

**INTRODUCTION**

Buses are an important part of public transport in places of the world which is available for all the people. There are buses owned by government and private limited companies. Bus travelling is a need for almost every citizen in a country.

The advantages of travelling by bus are – it is less stressful as one doesn’t need to worry about the traffic, travelling by bus is cheaper, reduces pollution, and one gets to meet and talk to new people.

But at the same time, the disadvantages of travelling by bus are – No seats available during long hour journeys, unhygienic maintenance, pickpocketing, theft, etc.

Due to growth of technology in a rapid speed, bus ticketing system has also come to the online platform.

Therefore, this project aims to book bus ticket(s) to safely travel to desired destinations. It is done by the emerging programming language: Python.

Member Functions:

1. Input\_data () – inputs data into mysql table – users\_details. Moves to LogIn\_win()
2. Next() – moves to Sign Up page 2

Member Functions:

1. SignUpWindow() – moves to SignUp\_wins()
2. SignInWindow() – moves to LogIn\_win()

**PROGRAM DESIGN**

Main Program

LogIn\_win()

After\_LogIn\_win()

Member Functions:

1. book() – Allows booking bus ticket(s)/hiring a bus by moving to Hire\_Book\_OP\_win().
2. edit() – Allows editing one’s user details by moving to Edit\_Pro\_win().
3. logout() – Pops a yes/no window to logout.

book()

Home\_win()

SignUp\_wins()

Member Functions:

1. LogIn() – checks whether the record of the user is in mysql table.

Mysql connectivity

Hire\_Book\_OP\_win()

Member Functions:

1. Validate2() – enables Book button.
2. HireBus() – Allows user to hire a bus.
3. BookTicket() - Allows user to book a bus ticket by moving to Book\_Tic\_win().
4. main()

Book\_Tic\_win ()

Member Functions:

1. Validate() – enables ‘Find Buses’ button.
2. Validate2() – enables Book button.
3. Find\_Bus() – Finds buses from table – Bus\_stand.

Edit\_Pro\_win()

Member Functions:

1. Editname() – Edits users’ username.
2. EditPassword() – Edits users’ password.
3. EditPhone() – Edits users’ phone number.
4. Editmail() - Edits users’ mail ID.

**PROGRAM CODE**

from tkinter import \*

from PIL import ImageTk,Image

import mysql.connector as sql

from tkinter import messagebox

import random, time

mydb = sql.connect(

host = 'localhost',

user = 'root',

passwd = 'P@ssword1',

charset = 'utf8',

database = 'mydatabase')

mycursor = mydb.cursor(buffered = True)

def Home\_win():

def SignUpWindow():

home\_window.destroy()

SignUp\_wins()

def SignInWindow():

home\_window.destroy()

LogIn\_win()

'''Windows along with canvas'''

# Home Window

home\_window = Tk()

home\_window.title('Home Page')

home\_window\_canvas = Canvas(home\_window, width=620, height=413)

home\_window\_canvas.grid(column=0,row=0)

home\_window\_image = ImageTk.PhotoImage(Image.open("Bus3.jpg"))

home\_window\_canvas.create\_image(25,125, image = home\_window\_image)

home\_window\_text = 'Welcome to Sify Bus Travel Agency! \n Pleased to have you here!'

home\_window\_message = Label(home\_window\_canvas, text = home\_window\_text, image = home\_window\_image, fg = 'black', font = ("Candara", 18, 'italic'), compound = 'bottom')#, justify = CENTER)

home\_window\_message.config(bg = 'light green')

home\_window\_message.grid(column = 1, row = 5, sticky = N)

'''Labels, Buttons and Entries'''

# home\_window:

Label1 = Label(home\_window, text = 'First time travelling in our agency?', font = ("Candara",15, 'italic'),background = 'orange', justify = LEFT)

Label1.grid(column = 30,row = 1)

home\_window\_canvas.create\_window(200,150, window = Label1)

Label2 = Label(home\_window, text = 'Travelled more than once?', font = ("Candara",15, 'italic'),background = 'orange', justify = LEFT)

Label2.grid(column = 30,row = 2)

home\_window\_canvas.create\_window(165,250, window = Label2)

Label3 = Label(home\_window, text = 'Remember: Social Distancing, Mask all the time, Sanitize!! :)', font = ("Candara",15, 'italic'),background = 'Plum', justify = LEFT)

Label3.grid(column = 500,row = 200)

home\_window\_canvas.create\_window(300,600, window = Label3)

SignUp = Button(home\_window, text = 'Sign Me Up!' , command = SignUpWindow, font = ("Candara",13, 'italic'), background = 'coral', fg = 'Black',anchor='w')

home\_window\_canvas.create\_window(450, 150, window=SignUp)

SignIn = Button(home\_window, text = 'Sign In' , command = SignInWindow, font = ("Candara",13, 'italic'), background = 'coral', fg = 'Black',anchor='w')

home\_window\_canvas.create\_window(430, 250, window=SignIn)

home\_window.geometry('764x641')#Right X Bottom

home\_window.resizable(False,False)#True,True)#

home\_window.mainloop()

def SignUp\_wins():

ID = random.randint(0,9999999)

print('Your user ID is : ', ID)

#Users\_details - User\_ID, First\_name, Last\_name, Gmail, Phone\_no, User\_name, Password

insert = 'Insert into users\_details(User\_ID, First\_name, Last\_name, Gmail, Phone\_no, User\_name, Password) values (%s,%s,%s,%s,%s,%s,%s)'

def Input\_data():

val1 = []

val1.extend(val)

val1.append(Username\_en.get())

val1.append(Password\_en.get())

print(val1)

tup = tuple(val1)

mycursor.execute(insert, tup)

mydb.commit()

print(mycursor.rowcount,"record(s) inserted")

SignUp\_window2.destroy()

LogIn\_win()

def Next():

global val

val = []

val.append(ID)

val.append(F\_Name\_en.get())

val.append(L\_Name\_en.get())

val.append(Gmail\_en.get())

val.append(doj\_en.get())

print(val)

SignUp\_window1.destroy()

global SignUp\_window2

SignUp\_window2 = Tk()

SignUp\_window2.title("Signing You Up! <Page 2>")

SignUp\_window2\_canvas = Canvas(SignUp\_window2, width=600, height=600)

SignUp\_window2\_canvas.grid(column=0,row=0)

SignUp\_window2\_image = ImageTk.PhotoImage(Image.open("Bus4.png"))

SignUp\_window2\_canvas.create\_image(350,300, image = SignUp\_window2\_image)#25,125

'''Labels, Entries, Button'''

Username = Label(SignUp\_window2, text = 'User Name', font = ('Candara',12), background = 'plum', justify = LEFT)

Username.grid(column = 0, row = 1)

SignUp\_window2\_canvas.create\_window(200,150, window = Username)

Password = Label(SignUp\_window2, text = 'New Password',font = ("Candara",12), background = 'plum', justify = LEFT)

Password.grid(column = 0, row = 2)

SignUp\_window2\_canvas.create\_window(200,190, window = Password)

ID\_dis = Label(SignUp\_window2, text = ' '+str(ID)+' : is your User ID,\n THIS IS A MUST DURING BOOKINGS',font = ("Candara",12), background = 'plum', justify = LEFT)

ID\_dis.grid(column = 0, row = 3)

SignUp\_window2\_canvas.create\_window(200,270, window = ID\_dis)

global Username\_en, Password\_en

Username\_en = Entry(SignUp\_window2)

Username\_en.grid(column = 1, row = 1)

SignUp\_window2\_canvas.create\_window(400,150, window = Username\_en)

Password\_en = Entry(SignUp\_window2, show='\*')

Password\_en.grid(column = 1, row = 2)

SignUp\_window2\_canvas.create\_window(400,190, window = Password\_en)

password = Label(SignUp\_window2, text = 'Password should atleast contain 8 characters & symbols',fg = 'White', font = ("Candara",10), background = 'black')

SignUp\_window2\_canvas.create\_window(300,230, window = password)

submit = Button(SignUp\_window2, text = 'Submit',command = Input\_data, font =("Candara",10), background = 'Sky blue')

submit.grid(column = 0, row = 3)

SignUp\_window2\_canvas.create\_window(200,320, window = submit)

SignUp\_window2.geometry('550x380')

SignUp\_window2.resizable(False,False)

SignUp\_window2.mainloop()

SignUp\_window2.destroy()

SignUp\_window1 = Tk()

SignUp\_window1.title('Signing you up - 1')

SignUp\_window1\_canvas = Canvas(SignUp\_window1, width=600, height=600)#620,413

SignUp\_window1\_canvas.grid(column=0,row=0)

SignUp\_window1\_image = ImageTk.PhotoImage(Image.open("Bus4.png"))

SignUp\_window1\_canvas.create\_image(350,300, image = SignUp\_window1\_image)#25,125

F\_Name = Label(SignUp\_window1, text = 'First Name', font = ("Candara",12), background = 'plum', justify = LEFT)

F\_Name.grid(column = 0,row = 1)

SignUp\_window1\_canvas.create\_window(200,150, window = F\_Name)

L\_Name = Label(SignUp\_window1, text = 'Last Name', font = ("Candara",12), background = 'plum', justify = LEFT)

L\_Name.grid(column = 0,row =2)

SignUp\_window1\_canvas.create\_window(200,190, window = L\_Name)

Gmail = Label(SignUp\_window1, text = 'Email',font = ("Candara",12), background = 'plum', justify = LEFT)

Gmail.grid(column = 0,row =3)

SignUp\_window1\_canvas.create\_window(200,230, window = Gmail)

doj = Label(SignUp\_window1, text = 'Phone number', font=("Candara",12), background = 'plum', justify = LEFT)

doj.grid(column = 0, row = 5)

SignUp\_window1\_canvas.create\_window(200,270, window = doj)

F\_Name\_en = Entry(SignUp\_window1)

F\_Name\_en.grid(column = 1,row=1)

SignUp\_window1\_canvas.create\_window(350,150, window = F\_Name\_en)

L\_Name\_en = Entry(SignUp\_window1)

L\_Name\_en.grid(column = 1,row=2)

SignUp\_window1\_canvas.create\_window(350,190, window = L\_Name\_en)

Gmail\_en = Entry(SignUp\_window1)

Gmail\_en.grid(column = 1,row=3)

SignUp\_window1\_canvas.create\_window(350,230, window = Gmail\_en)

doj\_en = Entry(SignUp\_window1)

doj\_en.grid(column = 1, row = 4)

SignUp\_window1\_canvas.create\_window(350,270, window = doj\_en)

Next = Button(SignUp\_window1, text = '<<<<< Next >>>>>', command = Next, font=("Candara",10), background = 'sky blue')

SignUp\_window1\_canvas.create\_window(200, 310, window = Next)

SignUp\_window1.geometry('600x450')

SignUp\_window1.resizable(False,False)

SignUp\_window1.mainloop()

SignUp\_window1.destroy()

def LogIn\_win():

## SignUp\_window2.destroy()

select = 'Select \* from users\_details where User\_Name = %s' #unique - primary key

select2 = 'Select \* from users\_details where Password = %s'

select2 = 'Select \* from users\_details where User\_ID = %d'

def LogIn():

try:

val = []

val.append(Username\_en.get())

val.append(Password\_en.get())

val.append(str(User\_ID\_en.get()))

tup = tuple(val)

print('Tuple',tup)

mycursor.execute(select, (tup[0],))

row = mycursor.fetchone()

print('Fetched row',row)

if (row[5] == Username\_en.get()) and (row[6] == Password\_en.get()) and (row[0] == int(User\_ID\_en.get())):

move = Label(LogIn\_window, text = ':) Moving to your account...')

LogIn\_window\_canvas.create\_window(400,350, window = move)

LogIn\_window.destroy()

After\_LogIn\_win()

#### elif ((row[0] != Username\_en.get()) and (row[1] == Password\_en.get())) or ((row[0] == Username\_en.get()) and (row[1] != Password\_en.get())):

elif (row[5] != Username\_en.get()) or (row[6] != Password\_en.get()) or (row[0] != int(User\_ID\_en.get())):

error = Label(LogIn\_window, text = '(^^)\' Sorry, either of the records are wrong.\n Please consider rechecking your inputs! ')

LogIn\_window\_canvas.create\_window(400,380, window = error)

elif (row[5] != Username\_en.get()) and (row[6] != Password\_en.get()) and (row[0] != int(User\_ID\_en.get())):

error = Label(LogIn\_window, text = ' :] Sorry, your acc does not exist, create one!')

LogIn\_window\_canvas.create\_window(400,380, window = error)

except TypeError:

error = Label(LogIn\_window, text = ' (^~^)\' Sorry, your acc does not exist, create one! ')

LogIn\_window\_canvas.create\_window(400,380, window = error)

LogIn\_window = Tk()

LogIn\_window.title('Log In')

LogIn\_window\_canvas = Canvas(LogIn\_window, width=600, height=600)

LogIn\_window\_canvas.grid(column=0,row=0)

LogIn\_window\_image = ImageTk.PhotoImage(Image.open("Bus5.png"))

LogIn\_window\_canvas.create\_image(350,300, image = LogIn\_window\_image)#25,125

'''Labels,Entries and Buttons'''

Username = Label(LogIn\_window, text = 'User Name',fg = 'White' , font = ('Candara',12), background = 'black', justify = LEFT)

LogIn\_window\_canvas.create\_window(200, 200, window = Username)

Password = Label(LogIn\_window, text = 'Password',fg = 'White' ,font = ("Candara",12), background = 'black', justify = LEFT)

LogIn\_window\_canvas.create\_window(200, 250, window = Password)

User\_ID = Label(LogIn\_window, text = 'User ID',fg = 'White' ,font = ("Candara",12), background = 'black', justify = LEFT)

LogIn\_window\_canvas.create\_window(200, 300, window = User\_ID)

Username\_en = Entry(LogIn\_window)

LogIn\_window\_canvas.create\_window(350, 200, window = Username\_en)

Password\_en = Entry(LogIn\_window, show = '\*')

LogIn\_window\_canvas.create\_window(350, 250, window = Password\_en)

User\_ID\_en = Entry(LogIn\_window)

LogIn\_window\_canvas.create\_window(350, 300, window = User\_ID\_en)

Log = Button(LogIn\_window, text = ' Log In ', command = LogIn, fg = 'black' , font = ('Candara',12), background = 'plum', justify = LEFT)

LogIn\_window\_canvas.create\_window(200, 350, window = Log)

LogIn\_window.geometry('600x450')

LogIn\_window.resizable(False,False)

LogIn\_window.mainloop()

def After\_LogIn\_win():

Home = Tk()

Home.title('Home page')

Home['background'] = 'Cyan'

Home\_canvas = Canvas(Home, width=620, height=413)

Home\_canvas.grid(column=0,row=0)

img = Image.open("1.png")

Home\_image = ImageTk.PhotoImage(img)

Home\_canvas.create\_image(25,125, image = Home\_image)

Home\_canvas.image = Home\_image

Home\_text = 'Welcome to your home page! '

Home\_message = Label(Home\_canvas, text = Home\_text, image = Home\_image, fg = 'black', font = ("Candara", 18, 'italic'), compound = 'bottom')#, justify = CENTER)

Home\_message.image = Home\_image

Home\_message.config(bg = 'sky blue')

Home\_message.grid(column = 1, row = 5, sticky = N)

def book():

Hire\_Book\_OP\_win()

def logout():

mess = messagebox.askyesno("Log out?", "Are you sure you want to log out?")

if mess:

Home.destroy()

Home\_win()

def edit():

Edit\_Pro\_win()

Book = Button(Home, text = 'Book a Ticket', command = book, fg = 'light green', background = 'black', font = ('Candara', 20))

Book.place(x = 30, y = 100)

hii = Button(Home, text = 'Hire a Bus', command = book, fg = 'light green', background = 'black', font = ('Candara', 20))

hii.place(x = 30, y = 200)

##

## Show = Button(Home, text = 'Show bookings', command = book, fg = 'light green', background = 'black', font = ('Candara', 20))

## Show.place(x = 900, y = 100)

Log\_out = Button(Home, text = 'Log out', command = logout, fg = 'light green', background = 'black', font = ('Candara', 20))

Log\_out.place(x = 900, y = 200)

Edit = Button(Home, text = 'Edit Profile', command = edit, fg = 'light green', background = 'black', font = ('Candara', 20))

Edit.place(x = 900, y = 100)

Home.geometry('1165x600')

Home.resizable(False,False)

Home.mainloop()

def Hire\_Book\_OP\_win():#hire/book options window

def Validate2():

print(r.get()+1)

print(rows[r.get()])

if r.get()+1:

Book['state'] = NORMAL

else:

messagebox.showerror("Error", "Please select the ID to book your ticket!")

def HireBus():

def ConfirmBooking():

fir = messagebox.askyesno("1st Conformation", "Confirm your booking?")

if fir:

sec = messagebox.askyesno("2nd Conformation", "Are you sure?")

if sec:

third = messagebox.showinfo("Thank you", "Thank You! Please screenshot your ticket displayed. \n\

Check Booked\_ticket.csv for your ticket details which would be saved TEMPORARILY")

book = Tk()

book.title('Hired Bus')

book['background'] = 'black'

# ID, Bus\_plate\_no, From\_dest, To\_dest, Phone\_no

index = r.get() # Booked row (bus)

di\_in = rows[index]

label1 = Label(book, text = ' Ticket Details ', font = ('Canadara', 18), fg = 'gold', background = 'black', relief = RAISED)

label1.place(x = 70, y = 10)

bus\_no = Label(book, text = 'Bus Number - ' + str(di\_in[1]),font = ('Canadara', 14), fg = 'white', background = 'black')

bus\_no.place(x = 20, y = 70)

bus\_type = Label(book, text = 'From destination - '+ str(di\_in[2]), font = ('Canadara', 14), fg = 'white', background = 'black')

bus\_type.place(x = 20, y = 100)

bus\_driver = Label(book, text = 'To destination - '+ str(di\_in[3]), font = ('Canadara', 14), fg = 'white', background = 'black')

bus\_driver.place(x = 20, y = 130)

frm = Label(book, text = 'Number of seats available - ' + str(45), font = ('Canadara', 14), fg = 'white', background = 'black')

frm.place(x = 20, y = 160)

to = Label(book, text = 'Phone no - '+ str(di\_in[4]), font = ('Canadara', 14), fg = 'white', background = 'black')

to.place(x = 20, y = 190)

book.geometry('550x350')

book.mainloop()

with open('Booked\_ticket.csv','w+') as f: ### NOT COMPLETED

f.truncate(0)

f.write('Bus Number - ' + str(di\_in[1]) + '\n')

f.write('From destination - '+ str(di\_in[2]) + '\n')

f.write('To destination - '+ str(di\_in[3]) + '\n')

f.write('Number of seats available - ' + str(45) + '\n')

f.write('Phone no - '+ str(di\_in[4]) + '\n')

Hire = Tk()

Hire.title('Buses available')

column\_names = ['ID', 'Bus Number', 'From Destination', 'To Destination', 'Phone number']

mycursor.execute('Select \* from Hire limit 0,10')

i = 0

flag = 0

global rows

rows = mycursor.fetchall()

for hire\_bus in rows:

for j in range(len(hire\_bus)):

e = Label(Hire, borderwidth = 1, relief = 'ridge', bg = 'cyan')

if flag == 0:

e.config(text = column\_names[j]+'\n'+str(hire\_bus[j]))

else:

e.config(text = str(hire\_bus[j]))

e.grid(row = i+1, column = j+1)

i += 1

flag = 1

global r, Book

pos\_y = 250

r = IntVar(Hire)

for k in range(len(rows)):

RB\_k = Radiobutton(Hire, text = str(rows[k][0]), variable = r, value = k)

RB\_k.place(x = 150, y = pos\_y)

pos\_y += 30

Back = Button(Hire, text = '<<<<< Back >>>>>', command = Hire.destroy,font = ("Candara", 10, 'italic'),background = 'Plum')

Back.place(x = i, y = 250)

Book = Button(Hire, text = 'Book',command = ConfirmBooking, fg = 'black', background = 'plum', font = ('Candara', 10))

Book.place(x = 200, y = 300)

Book['state'] = DISABLED

validate2 = Button(Hire, text = 'Validate <and click on book>',command = Validate2, fg = 'black', background = 'plum', font = ('Candara', 10))

validate2.place(x = 200, y = 250)

Hire.geometry('400x500')

Hire.mainloop()

def BookTicket():

Book\_Tic\_win()

def main():

global All\_Options

All\_Options = Tk()

All\_Options.title('Choice Options')

All\_Options['background'] = 'black'

message = Label(All\_Options, text = 'Here you can choose hire buses or \n book your ticket to travel between states.', fg = 'black', background = 'plum', font = ('candara', 18, 'italic'), compound = 'center')

message.place(x = 10, y = 20)

Hire = Button(All\_Options, text = 'To Hire bus(es)', command = HireBus, fg = 'black', background = 'cyan', font = ('Candara', 15))

Hire.place(x=20,y=120)

Book = Button(All\_Options, text = 'To Book your ticket', command = BookTicket, fg = 'black', background = 'cyan', font = ('Candara', 15))

Book.place(x=250,y=120)

All\_Options.geometry('500x320')

All\_Options.mainloop()

main()

def Book\_Tic\_win():

def Validate():

if From\_En.get() and To\_En.get() and Journ\_Date.get() and Journ\_Time.get():

Find['state'] = NORMAL

else:

messagebox.showerror("Error", "Please enter all the fields required!")

def Validate2():

print(r.get()+1)

print(rows[r.get()])

if r.get()+1:

Book['state'] = NORMAL

else:

messagebox.showerror("Error", "Please select the ID to book your ticket!")

def Find\_Bus():

print('Entered fields',From\_En.get(),To\_En.get(),Journ\_Date.get(),Journ\_Time.get())

if (From\_En.get() != None) and (To\_En.get() != None) and (Journ\_Date.get() != None) and (Journ\_Time.get() != None):

def ConfirmBooking():

fi = messagebox.askyesno("Confirmation", "Confirm your booking?")

if fi:

se = messagebox.askyesno("2nd confirmation", "Are you sure?")

if se:

th = messagebox.showinfo("Thank you", "Thank You! Please screenshot your ticket displayed. \n\

Check Booked\_ticket.csv for your ticket details which would be saved TEMPORARILY")

''' IMPORTANT '''

''' write the ticket details in a file '''

''' Create a table for ticket savings, and while connecting (all files to one) display

their name, ticket no, etc and also manage them using sql queries

Else - link users and bus\_stand'''

book = Tk()

book.title('Your ticket')

book['background'] = 'Black'

# ID 0, Bus\_No 1, From\_dest 2, From\_stand 3, To\_dest 4, To\_stand 5, departure\_timings 6

# Arrival\_timings 7 , Type 8, driver 9

index = r.get() # Booked row (bus)

di\_in = rows[index]

label1 = Label(book, text = ' Ticket Details ', font = ('Canadara', 18), fg = 'gold', background = 'black', relief = RAISED)

label1.place(x = 70, y = 10)

bus\_no = Label(book, text = 'Bus Number - ' + str(di\_in[1]),font = ('Canadara', 14), fg = 'white', background = 'black')

bus\_no.place(x = 20, y = 70)

bus\_type = Label(book, text = 'Bus type - '+ str(di\_in[8]), font = ('Canadara', 14), fg = 'white', background = 'black')

bus\_type.place(x = 20, y = 100)

bus\_driver = Label(book, text = 'Bus driver - '+ str(di\_in[9]), font = ('Canadara', 14), fg = 'white', background = 'black')

bus\_driver.place(x = 20, y = 130)

frm = Label(book, text = 'From destination - ' + str(di\_in[2]) + ' , ' + str(di\_in[3]), font = ('Canadara', 14), fg = 'white', background = 'black')

frm.place(x = 20, y = 160)

to = Label(book, text = 'To destination - '+ str(di\_in[4])+ ' , ' + str(di\_in[5]), font = ('Canadara', 14), fg = 'white', background = 'black')

to.place(x = 20, y = 190)

timed = Label(book, text = 'Timing of departure - '+ str(di\_in[6]), font = ('Canadara', 14), fg = 'white', background = 'black')

timed.place(x = 20, y = 220)

timea = Label(book, text = 'Timing of arrival - '+ str(di\_in[7]), font = ('Canadara', 14), fg = 'white', background = 'black')

timea.place(x = 20, y = 250)

fm = Label(book, text = 'No. of female(s) - '+ str(v.get()), font = ('Canadara', 14), fg = 'white', background = 'black')

fm.place(x = 20, y = 280)

m = Label(book, text = 'No. of male(s) - '+ str(v2.get()), font = ('Canadara', 14), fg = 'white', background = 'black')

m.place(x = 20, y = 310)

b = Label(book, text = 'No. of boy(s) - '+ str(v3.get()), font = ('Canadara', 14), fg = 'white', background = 'black')

b.place(x = 20, y = 340)

g = Label(book, text = 'No. of girl(s) - '+ str(v4.get()), font = ('Canadara', 14), fg = 'white', background = 'black')

g.place(x = 20, y = 370)

j = 0

seat = random.randint(1,45)

for i in range(v.get()+v2.get()+v3.get()+v3.get()):

sea = Label(book, text = 'Seat number - '+ str(seat+j), font = ('Canadara', 14), fg = 'white', background = 'black')

sea.place(x = 20, y = 430+j\*30)

j += 1

book.geometry('600x550')

with open('Booked\_ticket.csv','w+') as f: ### NOT COMPLETED

f.write('Bus Number - ' + str(di\_in[1]) + '\n')

f.write('Bus type - '+ str(di\_in[8]) + '\n')

f.write('Bus driver - '+ str(di\_in[9]) + '\n')

f.write('From destination - ' + str(di\_in[2]) + ' , ' + str(di\_in[3]) + '\n')

f.write('To destination - '+ str(di\_in[4])+ ' , ' + str(di\_in[5]) + '\n')

f.write('Timing of departure - '+ str(di\_in[6]) + '\n')

f.write('Timing of arrival - '+ str(di\_in[7]) + '\n')

f.write('No. of female(s) - '+ str(v.get()) + '\n')

f.write('No. of male(s) - '+ str(v2.get()) + '\n')

f.write('No. of boy(s) - '+ str(v3.get()) + '\n')

f.write('No. of girl(s) - '+ str(v4.get()) + '\n')

f.write('Journey Date - '+ str(Journ\_Date.get()) + '\n')

Booking2 = Tk()

Booking2.title('Available Buses')

Booking2['background'] = 'Cyan'

column\_names = ['ID', 'Bus Number', 'From Destination', 'Starting Point', 'To Destination', 'Ending Point',

'Departure Timings', 'Arrival Timings', 'Type', 'Driver']

mycursor.execute('Select \* from Bus\_Stand where From\_Dest = %s and To\_Dest = %s limit 0,10',\

(From\_En.get(), To\_En.get()))

flag = 0

pos\_y = 310

i = 0

global rows

rows = mycursor.fetchall()

for bus in rows:

for j in range(len(bus)):

e = Label(Booking2,borderwidth = 1,relief="ridge", bg = 'cyan')

if flag == 0:

e.config(text = column\_names[j]+'\n'+str(bus[j]))

else:

e.config(text = str(bus[j]))

e.grid(row = i+1, column = j+1)

# value - The value of each radiobutton is assigned to the control variable when it is turned on by the user.

# variable - It is the control variable which is used to keep track of the user's choices. It is shared among all the radiobuttons.

mess = Label(Booking2, text = 'Choose the corresponding ID of the row for which you want to book your bus ticket',fg = 'black', font = ('Candara', 15), background = 'sky blue')

mess.place(x = 250, y = 250)

## var = IntVar()

## Radio\_Buttoni = Radiobutton(Booking2, text = str(rows[i][0]), variable = var, value = 1, bg = 'light green')

## Radio\_Buttoni.place(x = 250, y = pos\_y)

flag = 1

i += 1

''' Radio Button '''

global r, Book

r = IntVar(Booking2)

for k in range(len(rows)):

RB\_k = Radiobutton(Booking2, text = str(rows[k][0]), variable = r, value = k)#rows[k][0])

RB\_k.place(x = 250, y = pos\_y)

pos\_y += 30

Back = Button(Booking2, text = '<<<<< Back >>>>>', command = Booking2.destroy,font = ("Candara", 10, 'italic'),background = 'Plum')

Back.place(x = i, y = 5\*50)

Book = Button(Booking2, text = 'Book',command = ConfirmBooking, fg = 'black', background = 'plum', font = ('Candara', 10))

Book.place(x = 500, y = 310)

Book['state'] = DISABLED

validate2 = Button(Booking2, text = 'Validate <and click on book>',command = Validate2, fg = 'black', background = 'plum', font = ('Candara', 10))

validate2.place(x = 300, y = 310)

Booking2.geometry('950x400')

Booking2.mainloop()

Booking = Tk()

Booking.title('Booking your ticket')

Booking['background'] = 'magenta'

Booking\_window\_text = '<<<<<<<< Book Your Ticket >>>>>>>>'

Booking\_window\_message = Label(Booking, text = Booking\_window\_text, fg = 'black', font = ("Candara", 18, 'italic'), compound = 'bottom')

Booking\_window\_message.config(bg = 'gold')

Booking\_window\_message.place(x = 200, y = 20)

'''Labels, Buttons'''

From = Label(Booking, text = 'From:', fg = 'black', font = ('Candara', 15), background = 'sky blue')

From.place(x = 100, y = 80)

clicked = StringVar()

clicked.set('Non-AC Semi-Sleeper')

Type = Label(Booking, text = 'Type:', fg = 'black', font = ('Candara', 15), background = 'sky blue')

Type.place(x = 540, y = 80)

Type\_Drop = OptionMenu(Booking, clicked, 'AC Sleeper','AC Semi-sleeper', 'Non-AC Sleeper', 'Non-AC Semi-Sleeper')

Type\_Drop.place(x = 600, y = 80)

To = Label(Booking, text = 'To : ', fg = 'black', font = ('Candara', 15), background = 'sky blue')

To.place(x = 100, y = 140)

Jour\_Date = Label(Booking, text = 'Journey Date: ', fg = 'black', font = ('Candara', 15), background = 'sky blue')

Jour\_Date.place(x = 100, y = 200)

Jour\_Time = Label(Booking, text = 'Journey Time : ', fg = 'black', font = ('Candara', 15), background = 'sky blue')

Jour\_Time.place(x = 100, y = 260)

Adult = Label(Booking, text = 'Adults: ', fg = 'black', font = ('Candara', 15), background = 'sky blue')

Adult.place(x = 100, y = 320)

Male = Label(Booking, text = 'Male: ', fg = 'black', font = ('Candara', 14), background = 'sky blue')

Male.place(x = 120, y = 380)

place\_x = 90#143

v = IntVar(Booking, '1')

Rbs = {'0':'0','1':'1','2':'2','3':'3','4':'4','5':'5','6':'6','7':'7','8':'8','9':'9','10':'10'}

for (key, value) in Rbs.items():

RB1 = Radiobutton(Booking, text = key, variable = v,

value = value, indicator = 0,

background = 'sky blue').place(x = place\_x+50, y = 420)#380

place\_x += 20

Female = Label(Booking, text = 'Female: ', fg = 'black', font = ('Candara', 14), background = 'sky blue')

Female.place(x = 380, y = 380)#120,445

v2 = IntVar(Booking, '1')

place\_x2 = 380#170

for (key, value) in Rbs.items():

RB2 = Radiobutton(Booking, text = key, variable = v2,

value = value, indicator = 0,

background = 'sky blue').place(x = place\_x2+50, y = 420)#450

place\_x2 += 20

Child = Label(Booking, text = 'Child(ren): ', fg = 'black', font = ('Candara', 15), background = 'sky blue')

Child.place(x = 100, y = 480)

Boy = Label(Booking, text = 'Boy: ', fg = 'black', font = ('Candara', 14), background = 'sky blue')

Boy.place(x = 120, y = 540)

v3 = IntVar(Booking, '1')

place\_x3 = 90

for (key, value) in Rbs.items():

RB3 = Radiobutton(Booking, text = key, variable = v3,

value = value, indicator = 0,

background = 'sky blue').place(x = place\_x3+50, y = 580)

place\_x3 += 20

Girl = Label(Booking, text = 'Girl: ', fg = 'black', font = ('Candara', 14), background = 'sky blue')

Girl.place(x = 380, y = 540)

v4 = IntVar(Booking, '1')

place\_x4 = 380

for (key, value) in Rbs.items():

RB4 = Radiobutton(Booking, text = key, variable = v4,

value = value, indicator = 0,

background = 'sky blue').place(x = place\_x4+50, y = 580)

place\_x4 += 20

'''Entries'''

From\_En = Entry(Booking)

From\_En.place(x = 200, y = 80)#, ipady = 3)

To\_En = Entry(Booking)

To\_En.place(x = 200, y = 140)

Journ\_Date = Entry(Booking)

Journ\_Date.place(x = 260, y = 200)

Journ\_Time = Entry(Booking)

Journ\_Time.place(x = 260, y = 260)

Find = Button(Booking, text = ' Find Buses ', command = Find\_Bus, font=('Candara',18))

Find['state'] = DISABLED

Find.place(x = 100, y = 630)

Validate = Button(Booking, text = 'Validate',command=Validate, font=('Candara',18))

Validate.place(x = 400, y = 630)

Booking.geometry('850x850')

Booking.mainloop()

def Edit\_Pro\_win():

def Editname():

Manage2 = Tk()

Manage2.title('Change username')

Manage2['background'] = 'plum'

name = Label(Manage2, text = 'New User Name', fg = 'white', background = 'black', font = ('Candara', 15))

name.place(x = 10, y = 50)

name\_en = Entry(Manage2)

name\_en.place(x = 200, y = 50)

password = Label(Manage2, text = 'Password', fg = 'white', background = 'black', font = ('Candara', 15))

password.place(x = 10, y = 100)

pass\_en = Entry(Manage2, show = '\*')

pass\_en.place(x = 200, y = 100)

BId = Label(Manage2, text = 'User ID', fg = 'white', background = 'black', font = ('Candara', 15))

BId.place(x = 10, y = 150)

bid\_en = Entry(Manage2)

bid\_en.place(x = 200, y = 150)

def update():

update1 = 'Update users\_details set user\_name = %s where password = %s and User\_ID = %s'

val = []

val.append(name\_en.get())

val.append(pass\_en.get())

val.append(bid\_en.get())

tup = tuple(val)

mycursor.execute(update1,tup)

mydb.commit()

print(mycursor.rowcount,"record(s) updated")

submit = Button(Manage2, text = 'Change', command = update, fg = 'white', background = 'black', font = ('Candara', 15))

submit.place(x = 120, y = 200)

Manage2.geometry('370x300')

Manage2.resizable(False,False)

Manage2.mainloop()

def EditPassword():

Manage2 = Tk()

Manage2.title('Change password')

Manage2['background'] = 'plum'

bid = Label(Manage2, text = 'User ID', fg = 'white', background = 'black', font = ('Candara', 15))

bid.place(x = 10, y = 50)

bid\_en = Entry(Manage2)

bid\_en.place(x = 250, y = 50)

password = Label(Manage2, text = 'New Password', fg = 'white', background = 'black', font = ('Candara', 15))

password.place(x = 10, y = 100)

pass\_en = Entry(Manage2, show = '\*')

pass\_en.place(x = 250, y = 100)

confirm = Label(Manage2, text = 'Please make sure that your password contains atleast 8 characters \n and atleast 3 special symbols', fg = 'black', background = 'cyan', font = ('Candara', 11))

confirm.place(x = 10, y = 150)

def update():

update = 'Update users\_details set password = %s where User\_ID = %s'

val = []

val.append(pass\_en.get())

val.append(bid\_en.get())

tup = tuple(val)

mycursor.execute(update,tup)

mydb.commit()

print(mycursor.rowcount,"record(s) updated")

submit = Button(Manage2, text = 'Change', command = update, fg = 'white', background = 'black', font = ('Candara', 15))

submit.place(x = 10, y = 200)

Manage2.geometry('480x300')

Manage2.resizable(False,False)

Manage2.mainloop()

def EditPhone():

Manage2 = Tk()

Manage2.title('Change Phone number')

Manage2['background'] = 'plum'

bid = Label(Manage2, text = 'User ID', fg = 'white', background = 'black', font = ('Candara', 15))

bid.place(x = 10, y = 50)

bid\_en = Entry(Manage2)

bid\_en.place(x = 250, y = 50)

phone = Label(Manage2, text = 'New Phone Number', fg = 'white', background = 'black', font = ('Candara', 15))

phone.place(x = 10, y = 100)

phone\_en = Entry(Manage2)

phone\_en.place(x = 250, y = 100)

def update():

# users\_personal - ID, first\_name, last\_name, gmail, user\_post, phone\_no

update = 'Update users\_details set Phone\_No = %s where user\_ID = %s'

val = []

val.append(phone\_en.get())

val.append(bid\_en.get())

tup = tuple(val)

mycursor.execute(update,tup)

mydb.commit()

print(mycursor.rowcount,"record(s) updated")

submit = Button(Manage2, text = 'Change', command = update, fg = 'white', background = 'black', font = ('Candara', 15))

submit.place(x = 10, y = 200)

Manage2.geometry('420x300')

Manage2.resizable(False,False)

Manage2.mainloop()

def Editmail():

Manage2 = Tk()

Manage2.title('Change Mail Address')

Manage2['background'] = 'plum'

bid = Label(Manage2, text = 'User ID', fg = 'white', background = 'black', font = ('Candara', 15))

bid.place(x = 10, y = 50)

bid\_en = Entry(Manage2)

bid\_en.place(x = 200, y = 50)

mail = Label(Manage2, text = 'New Email', fg = 'white', background = 'black', font = ('Candara', 15))

mail.place(x = 10, y = 100)

mail\_en = Entry(Manage2)

mail\_en.place(x = 200, y = 100)

def update():

# users\_personal - ID, first\_name, last\_name, gmail, user\_post, phone\_no

update = 'Update users\_details set gmail = %s where user\_ID = %s'

val = []

val.append(mail\_en.get())

val.append(bid\_en.get())

tup = tuple(val)

mycursor.execute(update,tup)

mydb.commit()

print(mycursor.rowcount,"record(s) updated")

submit = Button(Manage2, text = 'Change', command = update, fg = 'white', background = 'black', font = ('Candara', 15))

submit.place(x = 10, y = 200)

Manage2.geometry('370x300')

Manage2.resizable(False,False)

Manage2.mainloop()

Manage = Tk()

Manage.title('Manage your bookings')

Manage['background'] = 'Gold'

message = 'Options available for editing your profile'

label = Label(Manage, text = message, fg = 'black',background = 'sky blue', font = ('Candara', 18))#

label.place(x = 50, y = 10)

''' Buttons '''

Edit\_name = Button(Manage, text = '1. Edit your username', command = Editname, fg = 'white', background = 'black', font = ('Candara', 15))

Edit\_name.place(x = 50, y = 80)

Edit\_password = Button(Manage, text = '2. Edit your password', command = EditPassword, fg = 'white', background = 'black', font = ('Candara', 15))

Edit\_password.place(x = 50, y = 160)

Edit\_phone = Button(Manage, text = '3. Edit your phone number', command = EditPhone, fg = 'white', background = 'black', font = ('Candara', 15))

Edit\_phone.place(x = 50, y = 240)

Edit\_gmail = Button(Manage, text = '4. Edit your email', command = Editmail, fg = 'white', background = 'black', font = ('Candara', 15))

Edit\_gmail.place(x = 50, y = 320)

#Manage.resizable(False,False)

Manage.geometry('550x550')

Manage.mainloop()

Home\_win()

**SAMPLE OUTPUT**

1.



|  |  |
| --- | --- |
| After “Sign Me Up!” is clicked    2. (a) | Sample    2. (b) |
| 2. (c) | 2. (d) |

|  |  |
| --- | --- |
| 2. (e) | 2. (f) |

|  |  |
| --- | --- |
| After “Sign In” is clicked | Sample |
| 3. (a) | 3. (b) |



4.

|  |
| --- |
| After “Book a Ticket” is clicked |
| 5. (a) |

|  |
| --- |
| 5. (b) window displayed after “To Book your ticket” is clicked    5. (c) Sample    5. (d) Once “Validate” is clicked      5. (e) Once “Find Buses” is clicked    5. (f) Once “Validate <and click on book>” is clicked      5. (g) Once “Book” is clicked      5. (h)    5. (i) Ticket being saved in an excel sheet |

|  |
| --- |
| After “Hire a Bus” is clicked |
| 6. (a)    6. (b) After “To Hire bus(es)” is clicked        6. (c) After “Validate <and click on book>” is clicked    6. (d) After “Book” is clicked      6. (e)      6. (f) Ticket being saved in an excel sheet |

|  |  |
| --- | --- |
| After “Edit Profile” is clicked | |
| 7. (a) | |
| After “Edit your username” is clicked | Sample |
| 7. (a) | 7. (b) |

|  |  |
| --- | --- |
| After “Edit your password” is clicked | Sample |
| 7. (c) | 7. (d) |

|  |  |
| --- | --- |
| After “Edit your phone number” is clicked | Sample |
| 7. (e) | 7. (f) |

|  |  |
| --- | --- |
| After “Edit your email” is clicked | Sample |
| 7. (g) | 7. (h) |

|  |
| --- |
| After “Edit your email” is clicked |
| 8. (b)  8. (a) |

**BIBLIOGRAPHY**

* www.stackoverflow.com
* www.geeksforgeeks.org
* www.w3schools.com
* https://www.youtube.com/watch?v=YXPyB4XeYLA
* Computer Science textbook for class 12 by Sumita Arora
* Google Images