



School of Computer Science and Engineering

End-Term Report

Course Code: INT 213

Topic: Cab Booking System

Submitted by: Group -6

S.no.	Name	Roll No.	Registration no.
1	Abhijeet Kumar	50	11905729
2	Rishika Singh	05	11904961
3	Prakhar Srivastava	07	11904878

Submitted to: Ms Gagandeep Kaur

CONTENT

1. Objective

2. Introduction

3. Output Screenshot (GUI)

4. Source Code

5. Conclusion

6. Bibliography

1. Objective

The main objective of this project is to get user friendly with Python programming and Graphical User Interface (GUI). And to develop a Cab booking system interface with all the required functionalities.

This project also aims to provide a user-friendly interface to users to let them easily book their cab.

This cab booking system is easiest way to book cab to any location and that is what this project aims for.

There are different modules that we have provide in our interface such as main page which includes Sign-in option, Sign-up option. By clicking on Sign-in or Sign-up option you are redirected to another interface where you can sign-in or sign-up after that you are redirected to booking page where you can fill your required details and book your cab. There is a forgot interface also where you can change your password if you don't remember your password by entering your mobile number or Gmail. There is a interface for admin also where you can add a cab and search a cab.

In this it will easy for users to book their cab and for admin it will easy to add cab and search cab.

2. Introduction

Cab Booking management System is developed to manage all cab hiring work online. Using this system, it is very easy for customer to book a car online and car-booking agency can also view their booking online. So, it is also very useful for car booking agency. It is an online system through which customers can view available cabs, register the cabs, view profile and book cabs. This project intends to introduce more user friendly in the various activities such as record updating and searching.

Modules of the Projects:

- Welcome page module (Introduction to cab)
- Sign-in module
- Sign-up module
- Forgot password module
- Help Module
- Booking module
- Admin module

This project also includes SQL database connectivity that helps to record details and fetch it later, when required. Below given is the description of each module in project.

· **Welcome Module**

A welcome page is usually one or more web pages or modal overlays that appear the first time you open an app. The best welcome pages direct a user's focus to the welcome message, while also orienting them to the product.

“The content on your product welcome page should be consistent with what users experience elsewhere”

This module involves Sign-in option and Sign-up option.

If you click on sign-in option you will be forwarded to sign-in module and if you click on sign-up option you will be forwarded to sign-up module.

· **Sign-in Module**

A sign-in page is a web page or an entry page to a website that requires user identification and authentication, regularly performed by entering a g-mail and password combination.

Sign-in provide access to an entire site or part of a website. Signing in not only provides site access for the user, but also allows the website to track user actions and behaviour.

It includes a User entry and password for user or admin entry and password entry for admin. And there is a forgot password option so when you click on forgot password button you will be redirected to forgot password module.

· **Sign-up Module**

A signup page (also known as a registration page) enables users and organizations to independently register and gain access to your system. It is common to have multiple signup pages depending on the types of people and organizations you want to register.

You can signup using your phone number and g-mail with password. And your First name, Last name entries. After completing signup process, you will be forwarded to sign-in page.

· **Forgot Password Module**

Forgot password is the action of invalidating the current password for an account on a website, service, or device, and then creating a new one. A password may be reset using the settings of the software or service, or by contacting the customer service department.

It includes entries of mobile number or email, new password and confirm password.

· **Help Module**

A help page is a resource intended to provide the customer or end user with information and support related to a company's or institution's products and services. The purpose of a help desk is usually to troubleshoot problems or provide guidance about products such as computers, electronic equipment, food, apparel, or software.

It includes a drop-down selection where you can choose that where you are facing issue.

· **Booking Module**

Booking pages are the basis of schedule once's scheduling approach. They are pages through which bookings are made.

In booking page user can select different types of taxi that is City taxi, Outstation and rentals. After that you have to fill details like pickup location, drop location, when and type of taxi. Then you can confirm your

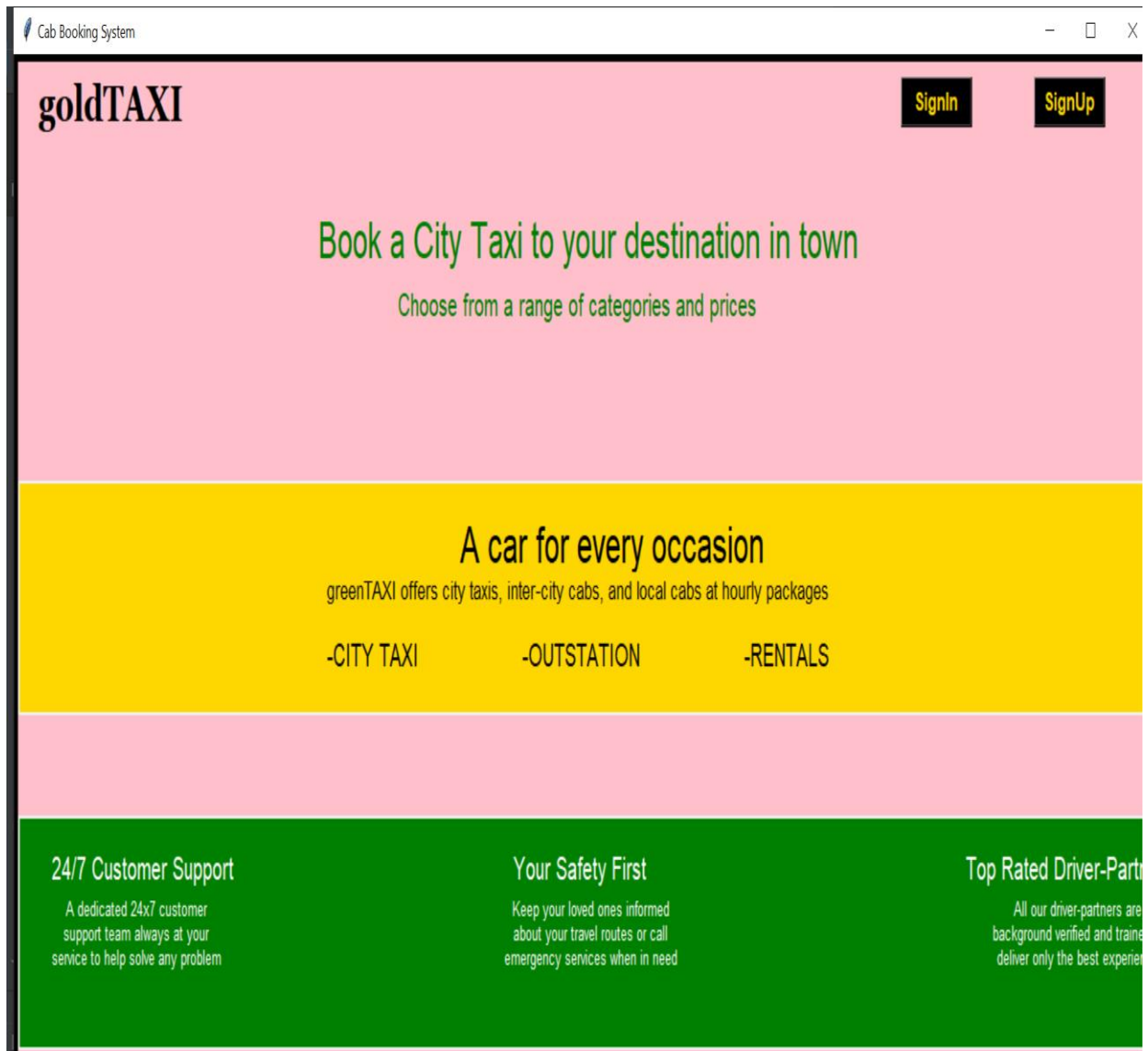
location and confirm your cab. Finally, you can confirm your booking.

· **Admin Module**

The Administration Panel (or the admin panel for short) is the primary tool for you to work with your online booking. Here you can manage your cab like details of cab, details of driver, view bookings and add cab.

3. Output Screenshots (GUI)

- Welcome Screen



• Sign-in Page



A screenshot of a web browser window displaying the 'goldTAXI' sign-in page. The browser's title bar reads 'Cab Booking System'. The page has a pink background. In the top left corner is the 'goldTAXI' logo. In the top right corner is a yellow banner with the text 'Get moving with goldTAXI'. Centered on the page is a yellow rectangular box with a black border containing the sign-in form. The form is titled 'SignIn' and includes two input fields: 'Email or mobile number' and 'Enter your password'. Below the password field is a blue link for 'Forgot Password?'. A black 'Next' button is positioned below the inputs. At the bottom of the box, it says 'Don't have an account? Sign Up'.

Cab Booking System

goldTAXI

Get moving with goldTAXI

SignIn

Email or mobile number

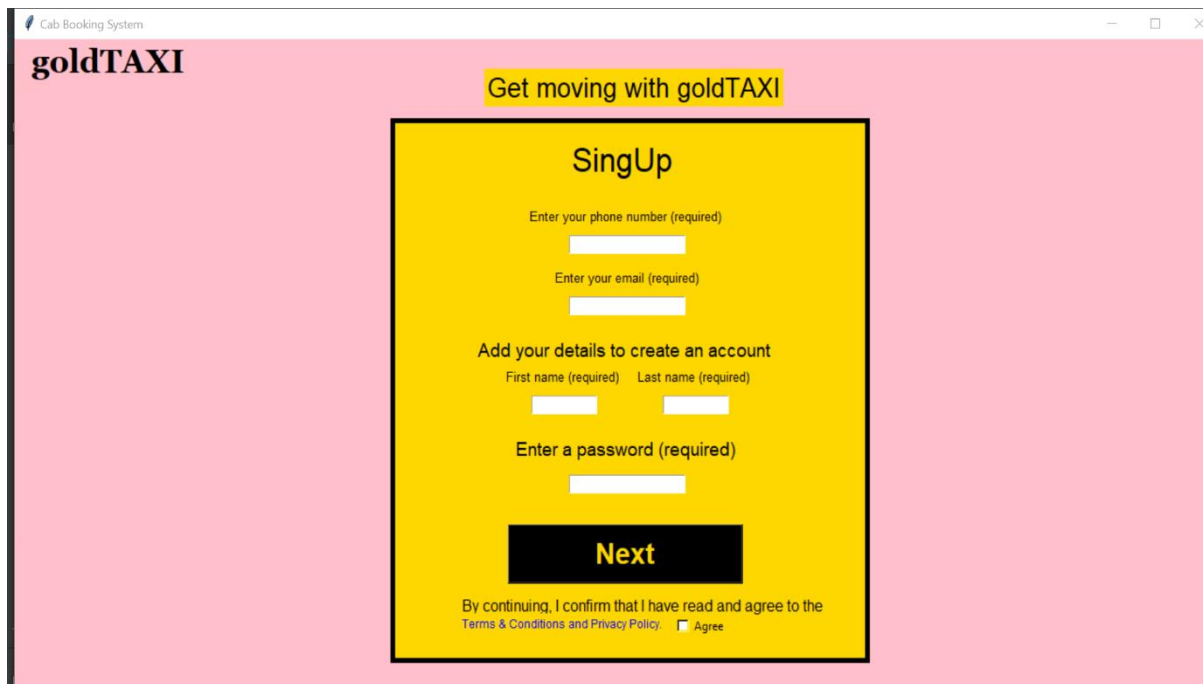
Enter your password

[Forgot Password?](#)

Next

Don't have an account? [Sign Up](#)

• Sign-up Page



A screenshot of a web browser window displaying the 'goldTAXI' sign-up page. The browser's title bar reads 'Cab Booking System'. The page has a pink background. In the top left corner is the 'goldTAXI' logo. In the top right corner is a yellow banner with the text 'Get moving with goldTAXI'. Centered on the page is a yellow rectangular box with a black border containing the sign-up form. The form is titled 'SingUp' and includes several input fields: 'Enter your phone number (required)', 'Enter your email (required)', 'First name (required)', 'Last name (required)', and 'Enter a password (required)'. Below the first and last name fields is a black 'Next' button. At the bottom of the box, there is a confirmation statement: 'By continuing, I confirm that I have read and agree to the Terms & Conditions and Privacy Policy.' followed by an unchecked checkbox and the word 'Agree'.

Cab Booking System

goldTAXI

Get moving with goldTAXI

SingUp

Enter your phone number (required)

Enter your email (required)

Add your details to create an account

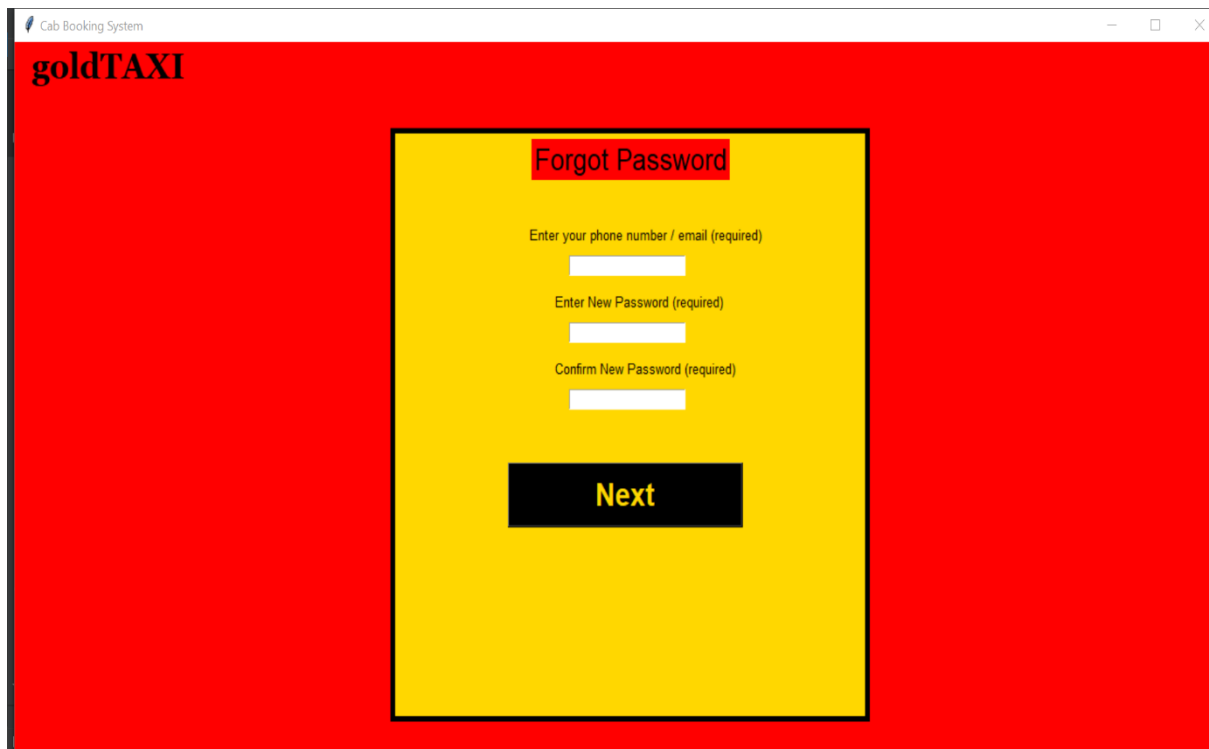
First name (required) Last name (required)

Enter a password (required)

Next

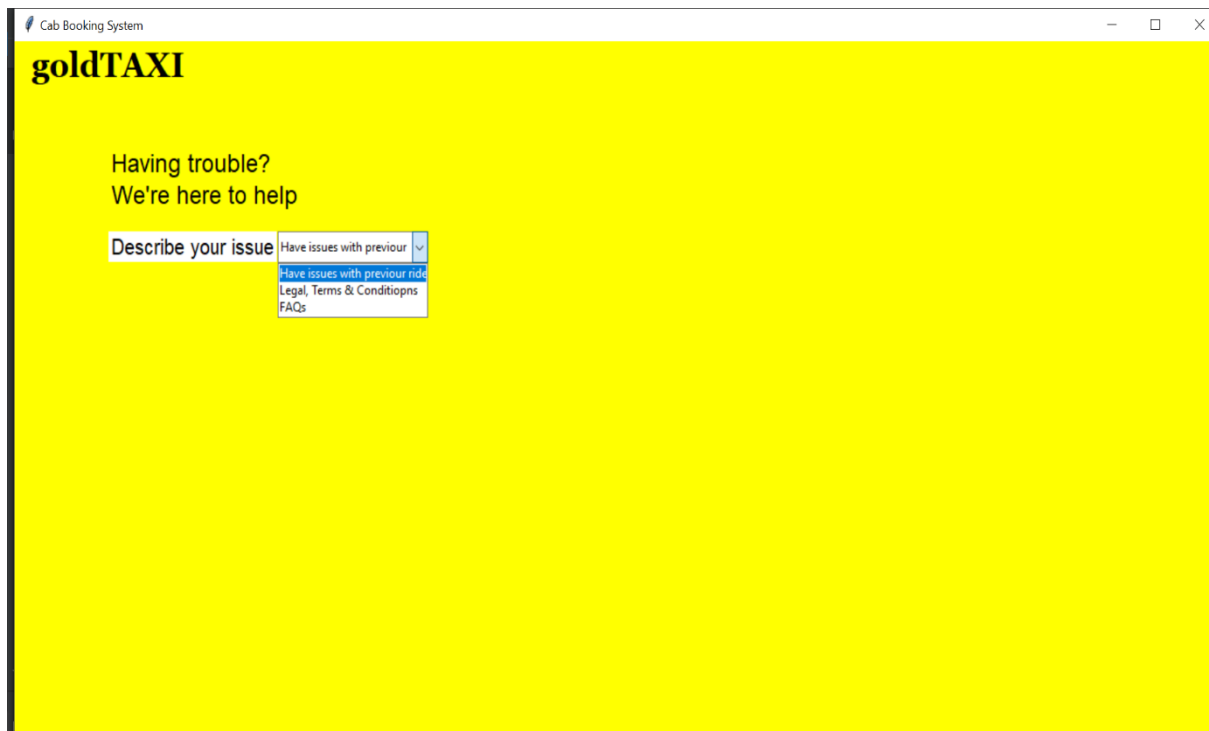
By continuing, I confirm that I have read and agree to the Terms & Conditions and Privacy Policy. ☐ Agree

• Forgot Password



The screenshot shows a web browser window titled "Cab Booking System". The page has a red background with the "goldTAXI" logo in the top left. In the center, there is a yellow rectangular box with a black border. Inside this box, the text "Forgot Password" is at the top. Below it, there are three input fields with labels: "Enter your phone number / email (required)", "Enter New Password (required)", and "Confirm New Password (required)". At the bottom of the yellow box is a black button with the word "Next" in yellow text.

• Help Page



The screenshot shows a web browser window titled "Cab Booking System". The page has a yellow background with the "goldTAXI" logo in the top left. Below the logo, the text "Having trouble? We're here to help" is displayed. Underneath, there is a text input field labeled "Describe your issue". To the right of this field is a dropdown menu with the text "Have issues with previous" and a downward arrow. The dropdown menu is open, showing three options: "Have issues with previous ride", "Legal, Terms & Conditions", and "FAQs".

- Booking Page (after signing as user)

Cab Booking System

goldTaxi

HELP

☒ CITY TAXI ☐ OUTSTATION ☐ RENTALS

PICKUP

DROP

WHEN

TYPE

Search Cab

pickup :Lajpat Nagar, Delhi
drop :Karol Bagh, Delhi
when :5Nov:2020

confirm

CAB DETAILS

car no	128
car type	mini
car name	ritz
car rate	300.0
car driver	vivek
car driver no	9260606040

- Admin Page (after signing as admin)

Cab Booking System

goldTAXI

Admin Account

HELP
ADD
SEARCH

car no
car type
car name
car rate
driver name
driver no
submit

car no
Search

car no
car type
car name
car rate
car driver
car driver no

Added Cab having car no 130

4. Source Code

Cab Booking System

"Tkinter is not the only GuiProgramming toolkit for Python."

"Tkinter is a Python binding to the Tk GUI toolkit."

"Import every exposed object in Tkinter into this current namespace."

"This System has total Seven Windows:" \

"window1 : Home Page : Constructor" \

"window2 : Sign In Page : SignIn Funtion" \

"window3 : Forgot Password : ForgotPassword Fuction

" \

"window4 : Admin's Account : portal1 Function" \

"window5 : User's Account : portal2 Function" \

"window6 : Help and Support : help Fuction" \

"window7 : Sign Up Page : SignUp Function" \

"admin : email - admin@gmail.com, pwd - 123" \

"user : email - user@gmail.com, pwd - 123" \

#importing required libraries

from tkinter import * # importing the tkinder
module and all the associated functions contained
within that module

from tkinter import ttk # provides access to the Tk
themed widget set

from tkinter import messagebox # module is used to
display message boxes

import webbrowser # provides a high-level
interface to allow displaying Web-based documents to
users

import sqlite3 # create a connection object
which will connect us to the database,

 # define tables, insert and change
rows, run queries and manage an SQLite database file
database name to be passed as parameter

conn = sqlite3.connect('data1base.db')

method to execute database commands

cur = conn.cursor()

empty list to later append the ids from the database

ids = []

created a class

class Cab:

 # defined a constructor

 def __init__(self,window): # Home Page of the
System

 # Declaration

 self.window1 = window

 self.window1.geometry("1350x750")

 self.window1.title("Cab Booking System")

 self.window1.configure(background="pink")

 self.email_or_phone = []

 self.password = []

 # ceated a frame

 self.frame1 = Frame(self.window1, height=690,
width=1350, highlightbackground="black",
highlightthickness=5, bg="pink")

 self.frame1.propagate(0)

 self.frame1.pack(side=TOP)

 # created label

 self.label1 = Label(self.window1, font=('georgia 25
bold'), text="goldTAXI", fg="black", bg="pink",
cursor="hand2") # direct to browser

```
self.label1.place(x=10, y=10, width=200)

self.label1.bind("<Button-1>", lambda e:
self.callback("http://www.google.com"))
```

```
# created buttons
```

```
self.button1 = Button(self.window1, font=("Arial",
12, "bold"), text="SignIn", bg="black",
fg="gold",command=self.SignIn) #direct to SignIn page
```

```
self.button1.place(x=1000, y=15, width=80)
```

```
self.button2 = Button(self.window1, font=("Arial",
12, "bold"), text="SignUp", bg="black",
fg="gold",command=self.SignUp) #direct to SignUp
page
```

```
self.button2.place(x=1150, y=15, width=80)
```

```
# created labels
```

```
self.label2 = Label(self.window1, font=("Arial", 25),
text="Book a City Taxi to your destination in town",
bg="pink",fg="green")
```

```
self.label2.place(x=340, y=100)
```

```
self.label3 = Label(self.window1, font=("Arial", 15),
text="Choose from a range of categories and prices",
bg="pink",fg="green")
```



```
self.label3.place(x=430, y=150)

# created canvas

self.canvas1 = Canvas(self.window1, bg='gold',
height=150, width=1335, cursor='pencil')

self.canvas1.place(x=5, y=280)

# created labels

self.label4 = Label(self.window1, font=("Arial", 25),
text="A car for every occasion", bg="gold", fg="black")

self.label4.place(x=500, y=300)

self.label5 = Label(self.window1, font=("Arial",
13),text="greenTAXI offers city taxis, inter-city cabs,
and local cabs at hourly packages",
bg="gold",fg="black")

self.label5.place(x=350, y=340)

self.label6 = Label(self.window1, font=("Arial", 15),
text="-CITY TAXI", bg="gold", fg="black")

self.label6.place(x=350, y=380)

self.label7 = Label(self.window1, font=("Arial", 15),
text="-OUTSTATION", bg="gold", fg="black")

self.label7.place(x=570, y=380)

self.label8 = Label(self.window1, font=("Arial", 15),
text="-RENTALS", bg="gold", fg="black")
```

```
self.label8.place(x=820, y=380)

# created canvas

self.canvas2 = Canvas(self.window1, bg='green',
height=150, width=1335, cursor='pencil')

self.canvas2.place(x=5, y=500)

# created labels

self.label8 = Label(self.window1, font=("Arial", 15),
text="24/7 Customer Support", bg="green",
fg="white")

self.label8.place(x=40, y=520)

self.label9 = Label(self.window1, font=("Arial",
10,),text="A dedicated 24x7 customer\nsupport team
always at your\nservice to help solve any
problem",bg="green", fg="white")

self.label9.place(x=40, y=550)

self.label10 = Label(self.window1, font=("Arial",
15), text="Your Safety First", bg="green", fg="white")

self.label10.place(x=560, y=520)

self.label11 = Label(self.window1, font=("Arial",
10),text="Keep your loved ones informed\nabout your
travel routes or call\nemergency services when in
need",bg="green", fg="white")

self.label11.place(x=550, y=550)
```

```
self.label12 = Label(self.window1, font=("Arial",  
15), text="Top Rated Driver-Partners", bg="green",  
fg="white")
```

```
self.label12.place(x=1070, y=520)
```

```
self.label13 = Label(self.window1, font=("Arial",  
10),text="All our driver-partners are\nbackground  
verified and trained to\ndeliver only the best  
experience",bg="green", fg="white")
```

```
self.label13.place(x=1100, y=550)
```

```
self.label14 = Label(self.window1, font=("Arial",  
10),text="Copyright © 2020 goldTAXI Pvt. Ltd. All rights  
reserved.", bg="pink", fg="black")
```

```
self.label14.place(x=940, y=660)
```

```
# function for web browser
```

```
def callback(self,url):    # direct to brower page  
    webbrowser.open_new(url)
```

```
# function for SignIn window
```

```
def SignIn(self):
```

```
# Declaration

self.window2 = Tk()

self.window2.geometry("1350x750")

self.window2.title("Cab Booking System")

self.window2.configure(background="pink")

# created frame

self.frame2 = Frame(self.window2, height=508,
width=500, highlightbackground="black",
highlightthickness=5, bg="gold")

self.frame2.propagate(0)

self.frame2.place(x=430,y=80)

# created labels

self.label1 = Label(self.window2, font=('georgia 25
bold'), text="goldTAXI", fg="black",
bg="pink",cursor="hand2") # direct to browser

self.label1.place(x=10, y=10, width=200)

self.label1.bind("<Button-1>", lambda e:
self.callback("http://www.google.com"))

self.label2 = Label(self.window2, font=("Arial", 20),
text="Get moving with goldTAXI", bg="gold",
fg="black")

self.label2.place(x=500, y=30)
```

```
self.label3 = Label(self.window2, font=("Arial", 30),
text="SignIn", bg="gold", fg="black")
self.label3.place(x=610, y=120)

self.label4 = Label(self.window2, font=("Arial", 8),
text="Email or mobile number", bg="gold", fg="black")
self.label4.place(x=600, y=180)

# created entry
self.entry1 = Entry(self.window2)
self.entry1.place(x=600, y=200,width=150)

# created label
self.label5 = Label(self.window2, font=("Arial", 8),
text="Enter your password", bg="gold", fg="black")
self.label5.place(x=600, y=230)

# created entry
self.entry2 = Entry(self.window2, show='*')
self.entry2.place(x=600, y=250, width=150)

# created label
self.label6 = Label(self.window2, font=("Arial", 8),
text="Forgot Password?", bg="gold",
fg="blue",cursor="hand2")
self.label6.place(x=660, y=270)
```

```
self.label6.bind("<Button-1>", lambda e:  
self.ForgotPassword()) # direct to forgot pwd  
window
```

```
# created button
```

```
self.button1 = Button(self.window2, font=("Arial",  
12, "bold"), text="Next", bg="black",  
fg="gold",command=self.condition) # verify  
credentials
```

```
self.button1.place(x=600, y=305, width=150)
```

```
# created labels
```

```
self.label7 = Label(self.window2, font=("Arial", 8),  
text="Don't have an account?", bg="gold", fg="black")
```

```
self.label7.place(x=600, y=335)
```

```
self.label8 = Label(self.window2, font=("Arial", 8),  
text="Sign Up", bg="gold", fg="blue",cursor="hand2")
```

```
self.label8.place(x=720, y=335)
```

```
self.label8.bind("<Button-1>", lambda e:  
self.SignUp()) # direct to SignUp page
```

```
# function to check credentials for SignIn
```

```
def condition(self):

    # All enties required
    if self.entry1.get() == "" or self.entry2.get() == "":
        messagebox.showerror("Error", "All fields
required")
        self.window2.destroy() # close SignIn page.

    # Admin Account Credentials
    elif self.entry1.get() == "admin@gmail.com" and
self.entry2.get() == "123":
        messagebox.showinfo("SignIn", f" :
{self.entry1.get()}")
        self.window2.destroy() # close SignIn page.
        self.portal1()      # redirect to admin account
interface.

    # User Account Credentials
    elif self.entry1.get() == "user@gmail.com" and
self.entry2.get() == "123":
        messagebox.showinfo("SignIn", f" :
{self.entry1.get()}")
```

```
self.window2.destroy() # close SignIn page.  
self.portal2()         # redirect to user account  
interface.
```

```
# In case Wrong Credentials  
else:  
    messagebox.showerror("Error", "Invalid  
Username or Password\n Try Again")  
    self.window2.destroy() # close SignIn page.
```

```
# function for forgot password  
def ForgotPassword(self):  
    # Declaration  
    self.window3 = Tk()  
    self.window3.geometry("1350x750")  
    self.window3.title("Cab Booking System")  
    self.window3.configure(background="Red")  
    # created frame  
    self.frame1 = Frame(self.window3, height=550,  
width=510, highlightbackground="black",  
highlightthickness=5, bg="gold")
```



```
self.frame1.propagate(0)
self.frame1.place(x=400, y=80)
# created labels
self.label1 = Label(self.window3, font=('georgia 25
bold'), text="goldTAXI", fg="black", bg="Red",
cursor="hand2") # direct to browser
self.label1.place(x=0, y=0, width=200)
self.label1.bind("<Button-1>", lambda e:
self.callback("http://www.google.com"))
self.label2 = Label(self.window3, font=("Arial", 20),
text="Forgot Password", bg="red", fg="black")
self.label2.place(x=550, y=90)
self.label3 = Label(self.window3, font=("Arial", 10),
text="Enter your phone number / email (required)",
bg="gold",fg="black")
self.label3.place(x=545, y=168)
# created entry
self.entry1 = Entry(self.window3)
self.entry1.place(x=590, y=198)
# created labels
```

```
self.label4 = Label(self.window3, font=("Arial", 10),  
text="Enter New Password (required)", bg="gold",  
fg="black")
```

```
self.label4.place(x=572, y=230)
```

```
# created entry
```

```
self.entry2 = Entry(self.window3)
```

```
self.entry2.place(x=590, y=260)
```

```
# created label
```

```
self.label5 = Label(self.window3, font=("Arial", 10),  
text="Confirm New Password (required)",  
bg="gold",fg="black")
```

```
self.label5.place(x=572, y=292)
```

```
# created entry
```

```
self.entry3 = Entry(self.window3)
```

```
self.entry3.place(x=590, y=322)
```

```
# created button
```

```
self.mail = Button(self.window3, font=("Arial", 22,  
"bold"), text="Next", bg="black",  
fg="gold",command=self.mail) # fuction for message
```

```
self.mail.place(x=525, y=390, width=250)
```

```
# function for mail, in case needs any help and support.
```

```
def mail(self):  
    messagebox.showinfo("Help & Support", "Email  
send to goldTaxi team")
```

```
# funtion for admin account
```

```
def portal1(self):  
    # Declaration  
    self.window4 = Tk()  
    self.window4.geometry("1350x750")  
    self.window4.title("Cab Booking System")  
    self.window4.configure(background="pink")  
    # created labels  
    self.label1 = Label(self.window4, font=('georgia 25  
bold'), text="goldTAXI", fg="black",  
bg="pink",cursor="hand2") # direct to browser  
    self.label1.place(x=10, y=10, width=200)  
    self.label1.bind("<Button-1>", lambda e:  
self.callback("http://www.google.com"))
```

```
self.label2=Label(self.window4,font=('arial
18'),text="Admin Account",fg="red",bg="pink" )
self.label2.place(x=20,y=80)
# created buttons
self.button1 = Button(self.window4, font=("Arial",
15), text="HELP", bg="steelblue", fg="gold",
command=self.help) # direct to help function
self.button1.place(x=30, y=150)
self.button2 = Button(self.window4, font=("Arial",
15), text="ADD", bg="steelblue", fg="gold",
command=self.add_cab) # direct to add function
self.button2.place(x=30, y=190)
self.button3 = Button(self.window4, font=("Arial",
15), text="SEARCH", bg="steelblue", fg="gold",
command=self.search_cab) # direct to search function
self.button3.place(x=30, y=230)
```

#function to add cab details by admin

```
def add_cab (self):
```

```
    # created label
```

```
self.car_no = Label(self.window4, text="car no",
font=('georgia 12'), fg='black', bg='pink')
self.car_no.place(x=300, y=200)
self.car_type = Label(self.window4, text="car
type", font=('georgia 12'), fg='black', bg='pink')
self.car_type.place(x=300, y=220)
self.car_name = Label(self.window4, text="car
name", font=('georgia 12'), fg='black', bg='pink')
self.car_name.place(x=300, y=240)
self.car_rate = Label(self.window4, text="car rate",
font=('georgia 12'), fg='black', bg='pink')
self.car_rate.place(x=300, y=260)
self.car_driver = Label(self.window4, text="driver
name", font=('georgia 12'), fg='black', bg='pink')
self.car_driver.place(x=300, y=280)
self.car_driverno = Label(self.window4,
text="driver no", font=('georgia 12'), fg='black',
bg='pink')
self.car_driverno.place(x=300, y=300)
# Entries for all labels
self.car_no_ent = Entry(self.window4, width=30)
self.car_no_ent.place(x=500, y=200)
```

```
self.car_type_ent = Entry(self.window4, width=30)
self.car_type_ent.place(x=500, y=220)
self.car_name_ent = Entry(self.window4,
width=30)
self.car_name_ent.place(x=500, y=240)
self.car_rate_ent = Entry(self.window4, width=30)
self.car_rate_ent.place(x=500, y=260)
self.car_driver_ent = Entry(self.window4,
width=30)
self.car_driver_ent.place(x=500, y=280)
self.car_driverno_ent = Entry(self.window4,
width=30)
self.car_driverno_ent.place(x=500, y=300)
# button to perform a command
self.submit = Button(self.window4, text="submit",
width=25, height=1, bg='steelblue',
command=self.add_cab_details) # call fuction to add
data
self.submit.place(x=500, y=330)

# getting the number of appointments fixed to
view in the log
```

```
sql2 = "SELECT car_no FROM avi "      #table name
'avi' in database 'data1base.db'

self.result = cur.execute(sql2)

for self.row in self.result:
    self.id = self.row[0]
    ids.append(self.id)

# ordering the ids
self.new = sorted(ids)
self.final_id = self.new[len(ids) - 1]

# displaying the logs in our right frame
self.box = Text(self.window4, width=50, height=5)
self.box.place(x=800, y=200)
```

funtion to call when the submit button is clicked

```
def add_cab_details(self):
    # getting the user inputs
    self.val1 = self.car_no_ent.get()
    self.val2 = self.car_type_ent.get()
    self.val3 = self.car_name_ent.get()
    self.val4 = self.car_rate_ent.get()
```

```

self.val5 = self.car_driver_ent.get()
self.val6 = self.car_driverno_ent.get()
# checking if the user input is empty
if self.val1 == " " or self.val2 == " ":
    messagebox.showinfo("Warning", "Please Fill Up
All Boxes")
else:
    # now we add to the database
    sql = "INSERT INTO 'avi'
(car_no,car_type,car_name,car_rate,car_driver,
car_driverno) VALUES(?, ?, ?,?, ?, ?)"    #table name
'avi' in database 'data1base.db'

    cur.execute(sql, (self.val1, self.val2, self.val3,
self.val4, self.val5, self.val6))
    conn.commit()

    messagebox.showinfo("Success", "car type " +
str(self.val2) + " avalaible")

    self.box.insert(END, 'Added Cab having car no ' +
str(self.val1))

# function for search cab details by admin
def search_cab(self):

```



```

# created label
self.car_no = Label(self.window4, text="car no",
font=('arial 12'), bg="pink", fg="black")
self.car_no.place(x=300, y=380)
# ceated entry
self.car_no_ent = Entry(self.window4, width=30)
self.car_no_ent.place(x=500, y=380)
# created search button
self.search = Button(self.window4, text="Search",
width=25, height=1, bg='steelblue',
command=self.search_cab_details) # fuction to search
data
self.search.place(x=500, y=410)

# function to search in database
def search_cab_details(self):
    self.input = self.car_no_ent.get()
    # execute sql commands
    sql = "SELECT * FROM avi WHERE car_no LIKE ?"
#table name 'avi' in database 'data1base.db'
    self.res = cur.execute(sql, (self.input,))
    for self.row in self.res:

```

```
self.car_no = self.row[1]
self.car_type = self.row[2]
self.car_name = self.row[3]
self.car_rate = self.row[4]
self.car_driver = self.row[5]
self.car_driverno = self.row[6]
```

```
# created labels
```

```
self.car_no1 = Label(self.window4, text="car no",
font=('arial 12'), fg='black', bg='pink')
self.car_no1.place(x=300, y=440)
self.car_type1 = Label(self.window4, text="car
type", font=('arial 12'), fg='black', bg='pink')
self.car_type1.place(x=300, y=460)
self.car_name1 = Label(self.window4, text="car
name", font=('arial 12'), fg='black', bg='pink')
self.car_name1.place(x=300, y=480)
self.car_rate1 = Label(self.window4, text="car
rate", font=('arial 12'), fg='black', bg='pink')
self.car_rate1.place(x=300, y=500)
self.car_driver1 = Label(self.window4, text="car
driver", font=('arial 12'), fg='black', bg='pink')
```

```
self.car_driver1.place(x=300, y=520)
self.car_driverno1 = Label(self.window4, text="car
driver no", font=('arial 12'), fg='black', bg='pink')
self.car_driverno1.place(x=300, y=540)

# entries for each labels
# filling the search result in the entry box to
update
self.ent1 = Entry(self.window4, width=30)
self.ent1.place(x=500, y=440)
self.ent1.insert(END, str(self.car_no))
self.ent2 = Entry(self.window4, width=30)
self.ent2.place(x=500, y=460)
self.ent2.insert(END, str(self.car_type))
self.ent3 = Entry(self.window4, width=30)
self.ent3.place(x=500, y=480)
self.ent3.insert(END, str(self.car_name))
self.ent4 = Entry(self.window4, width=30)
self.ent4.place(x=500, y=500)
self.ent4.insert(END, str(self.car_rate))
self.ent5 = Entry(self.window4, width=30)
```

```
self.ent5.place(x=500, y=520)
self.ent5.insert(END, str(self.car_driver))
self.ent6 = Entry(self.window4, width=30)
self.ent6.place(x=500, y=540)
self.ent6.insert(END, str(self.car_driverno))
```

funtion for users account page

```
def portal2(self):
```

```
    # Declaration
```

```
    self.window5 = Tk()
```

```
    self.window5.geometry("1350x750")
```

```
    self.window5.title("Cab Booking System")
```

```
    self.window5.configure(background="pink")
```

```
    #created frame
```

```
    self.frame2 = Frame(self.window5, height=520,
width=500, highlightbackground="black",
highlightthickness=5,bg="gold")
```

```
    self.frame2.propagate(0)
```

```
    self.frame2.place(x=50, y=100)
```

```
    # label for the window
```

```
self.heading = Label(self.window5,  
text="goldTaxi", font=('georgia 40 bold'), fg='black',  
bg='gold') # direct to browser
```

```
self.heading.place(x=180, y=30)
```

```
self.heading.bind("<Button-1>", lambda e:  
self.callback("http://www.google.com"))
```

```
# button for help & support
```

```
self.button1 = Button(self.window5, font=("Arial",  
15), text="HELP", bg="black",  
fg="gold", command=self.help) # direct to help and  
support window
```

```
self.button1.place(x=1200, y=20)
```

```
# checkbutton to select options of ride
```

```
self.checkbutton1=Checkbutton(self.window5,  
text="CITY TAXI", bg="gold")
```

```
self.checkbutton1.place(x=130,y=200)
```

```
self.checkbutton2=Checkbutton(self.window5,  
text="OUTSTATION", bg="gold")
```

```
self.checkbutton2.place(x=250, y=200)
```

```
self.checkbutton3=Checkbutton(self.window5,  
text="RENTALS", bg="gold")
```

```
self.checkbutton3.place(x=390, y=200)

# created labels

self.pickup = Label(self.window5, text="PICKUP",
font=('georgia 16 bold'), fg='black', bg='gold')
self.pickup.place(x=130, y=300)

self.drop = Label(self.window5, text="DROP",
font=('georgia 16 bold'), fg='black', bg='gold')
self.drop.place(x=130, y=340)

self.when = Label(self.window5, text="WHEN",
font=('georgia 16 bold'), fg='black', bg='gold')
self.when.place(x=130, y=380)

self.cab_type = Label(self.window5, text="TYPE",
font=('georgia 16 bold'), fg='black', bg='gold')
self.cab_type.place(x=130, y=420)

# Entries for all labels

self.pickup_ent = Entry(self.window5, width=30)
self.pickup_ent.place(x=280, y=300)

self.drop_ent = Entry(self.window5, width=30)
self.drop_ent.place(x=280, y=340)

self.when_ent = Entry(self.window5, width=30)
self.when_ent.place(x=280, y=380)

self.cab_type = Entry(self.window5, width=30)
```

```
self.cab_type.place(x=280, y=420)
```

```
# button to perform a command
```

```
self.search = Button(self.window5,font=("Arial",  
12, "bold"),text="Search  
Cab",width=30,height=2,fg="gold",bg='black',comman  
d=self.confirm_booking) #function to book process
```

```
self.search.place(x=145, y=460)
```

```
#function for showing details
```

```
def confirm_booking(self):
```

```
    x = self.pickup_ent.get()
```

```
    y = self.drop_ent.get()
```

```
    z = self.when_ent.get()
```

```
    # creating box and labes in it.
```

```
    self.box = Text(self.window5, width=50, height=7)
```

```
    self.box.place(x=800, y=200)
```

```
    label1 = Label(self.window5,font=("Arial", 12,  
"bold"),bg='white',text="pickup : " +  
x).place(x=810,y=210)
```

```
label2 = Label(self.window5,font=("Arial", 12,
"bold"),bg='white', text="drop : " + y).place(x=810,
y=230)
```

```
label3 = Label(self.window5, font=("Arial", 12,
"bold"),bg='white',text="when : " + z).place(x=810,
y=250)
```

```
label4 = Label(self.window5,font=("Arial", 13,
"bold"),bg='white', text="verify details").place(x=810,
y=290)
```

```
self.button3 = Button(self.window5, font=("Arial",
15), text="confirm", bg="steelblue",
fg="gold",command=self.show_cab_available) #
function to display data
```

```
self.button3.place(x=800, y=330)
```

```
# function for displaying cab details at the time of
booking.
```

```
def show_cab_available(self):
```

```
self.input = self.cab_type.get()
```

```
# execute sql commands
```



```
sql = "SELECT * FROM avi WHERE car_type LIKE ?"
#table name 'avi' in database 'data1base.db'
self.res = cur.execute(sql, (self.input,))
for self.row in self.res:
    self.car_no = self.row[1]
    self.car_type = self.row[2]
    self.car_name = self.row[3]
    self.car_rate = self.row[4]
    self.car_driver = self.row[5]
    self.car_driverno = self.row[6]
# create labels
self.label1 = Label(self.window5, text="AVAILABLE
CAB DETAILS", font=('arial 12'), fg='black', bg='pink')
self.label1.place(x=600, y=420)
self.car_no1 = Label(self.window5, text="car no",
font=('arial 12'), fg='black', bg='pink')
self.car_no1.place(x=600, y=440)
self.car_type1 = Label(self.window5, text="car
type", font=('arial 12'), fg='black', bg='pink')
self.car_type1.place(x=600, y=460)
self.car_name1 = Label(self.window5, text="car
name", font=('arial 12'), fg='black', bg='pink')
```

```
self.car_name1.place(x=600, y=480)
self.car_rate1 = Label(self.window5, text="car
rate", font=('arial 12'), fg='black', bg='pink')
self.car_rate1.place(x=600, y=500)
self.car_driver1 = Label(self.window5, text="car
driver", font=('arial 12'), fg='black', bg='pink')
self.car_driver1.place(x=600, y=520)
self.car_driverno1 = Label(self.window5, text="car
driver no", font=('arial 12'), fg='black', bg='pink')
self.car_driverno1.place(x=600, y=540)
```

entries for each labels

filling the search result in the entry box to
update

```
self.ent1 = Entry(self.window5, width=30)
self.ent1.place(x=800, y=440)
self.ent1.insert(END, str(self.car_no))
self.ent2 = Entry(self.window5, width=30)
self.ent2.place(x=800, y=460)
self.ent2.insert(END, str(self.car_type))
self.ent3 = Entry(self.window5, width=30)
self.ent3.place(x=800, y=480)
```

```
self.ent3.insert(END, str(self.car_name))
self.ent4 = Entry(self.window5, width=30)
self.ent4.place(x=800, y=500)
self.ent4.insert(END, str(self.car_rate))
self.ent5 = Entry(self.window5, width=30)
self.ent5.place(x=800, y=520)
self.ent5.insert(END, str(self.car_driver))
self.ent6 = Entry(self.window5, width=30)
self.ent6.place(x=800, y=540)
self.ent6.insert(END, str(self.car_driverno))

self.cabBook = Button(self.window5, font=("Arial",
12, "bold"), text="Cab Book", width=18, height=2,
fg="gold",bg='black', command=self.cab_booked)
#function for message

self.cabBook.place(x=800, y=560)


# function called after confirming booking.
def cab_booked(self):

    messagebox.showinfo("Cab Booked", "Email send
to your registered email\nThank you for using our
service.")
```

```
# funtion for help and support window
def help(self):
    # Declaration
    self.window6 = Tk()
    self.window6.geometry("1350x750")
    self.window6.title("Cab Booking System")
    self.window6.configure(background="yellow")
    # created labels
    self.label1 = Label(self.window6, font=('georgia 25
bold'), text="goldTAXI", fg="black",
bg="yellow", cursor="hand2") # direct to browser
    self.label1.place(x=0, y=0, width=200)
    self.label1.bind("<Button-1>", lambda e:
self.callback("http://www.google.com"))
    self.label2 = Label(self.window6, font=("Arial", 18),
text="Having trouble?", bg="yellow", fg="black")
    self.label2.place(x=100, y=100)
    self.label3 = Label(self.window6, font=("Arial", 18),
text="We're here to help", bg="yellow", fg="black")
    self.label3.place(x=100, y=130)
```

```
self.label4 = Label(self.window6, font=("Arial", 15),  
text="Describe your issue", bg="white", fg="black")
```

```
self.label4.place(x=100, y=180)
```

```
# created combobox
```

```
self.cur = ["Have issues with previouir ride", "Legal,  
Terms & Conditiopns","FAQs"]
```

```
self.cb = ttk.Combobox(self.window6,  
values=self.cur, width=10)
```

```
self.cb.place(x=280, y=180, width=160, height=30)
```

```
self.cb.current(0)
```

```
self.support = Button(self.window6, font=("Arial",  
12, "bold"), text="Click here", width=15, height=1,  
bg='steelblue', command=self.supportMsg) # function  
for supoort message
```

```
self.support.place(x=280, y=280)
```

```
# function called after support & help asked .
```

```
def supportMsg(self):
```

```
    messagebox.showinfo("mailed!", "Our team will  
contact you sortly.\nThank you for using our service.")
```

```
# funtion for SignUp Window
```

```
def SignUp(self):  
    # Declaration  
    self.window7 =Tk()  
    self.window7.geometry("1350x750")  
    self.window7.title("Cab Booking System")  
    self.window7.configure(background="pink")  
    # created frame  
    self.frame1 = Frame(self.window7, height=550,  
width=510, highlightbackground="black",  
highlightthickness=5, bg="gold")  
    self.frame1.propagate(0)  
    self.frame1.place(x=400, y=80)  
    # created labels  
    self.label1 = Label(self.window7, font=('georgia 25  
bold'), text="goldTAXI", fg="black",  
bg="pink",cursor="hand2") # direct to browser  
    self.label1.place(x=0, y=0, width=200)  
    self.label1.bind("<Button-1>", lambda e:  
self.callback("http://www.google.com"))
```

```
self.label11 = Label(self.window7, font=("Arial",
20), text="Get moving with goldTAXI", bg="gold",
fg="black")

self.label11.place(x=500, y=30)

self.label2 = Label(self.window7, font=("Arial", 25),
text="SingUp", bg="gold", fg="black")

self.label2.place(x=590, y=100)

self.label3
=Label(self.window7,font=("Arial",10),text="Enter your
phone number (required)",bg="gold",fg="black")

self.label3.place(x=545,y=168)

# created entry

self.entry1 = Entry(self.window7)

self.entry1.place(x=590,y=198)

# created label

self.label4 = Label(self.window7, font=("Arial", 10),
text="Enter your email (required)", bg="gold",
fg="black")

self.label4.place(x=572, y=230)

# created entry

self.entry2 = Entry(self.window7)

self.entry2.place(x=590, y=260)
```

```
# created labels

self.label5 = Label(self.window7, font=("Arial", 14),
text="Add your details to create an account",
bg="gold", fg="black")

self.label5.place(x=490, y=300)

self.label6 = Label(self.window7, font=("Arial", 10),
text="First name (required)", bg="gold", fg="black")

self.label6.place(x=520, y=330)

# created entry

self.entry3 = Entry(self.window7)

self.entry3.place(x=550, y=360,width=70)

# created label

self.label7 = Label(self.window7, font=("Arial", 10),
text="Last name (required)", bg="gold", fg="black")

self.label7.place(x=660, y=330)

# created entry

self.entry4 = Entry(self.window7)

self.entry4.place(x=690, y=360,width=70)

# created label

self.label8 = Label(self.window7, font=("Arial", 14),
text="Enter a password (required)", bg="gold",
fg="black")
```



```
self.label8.place(x=530, y=400)
```

```
# created entry
```

```
self.entry5 = Entry(self.window7)
```

```
self.entry5.place(x=590, y=440)
```

```
# created button
```

```
self.button3 = Button(self.window7, font=("Arial",  
22, "bold"), text="Next", bg="black",  
fg="gold", command=self.SignIn) #direct to SignIn
```

```
self.button3.place(x=525, y=490, width=250)
```

```
# created labels
```

```
self.label9 = Label(self.window7, font=("Arial", 12),  
text="By continuing, I confirm that I have read and  
agree to the", bg="gold", fg="black")
```

```
self.label9.place(x=473, y=560)
```

```
self.label10 = Label(self.window7, font=("Arial", 9),  
text="Terms & Conditions and Privacy Policy.",  
bg="gold", fg="blue")
```

```
self.label10.place(x=473, y=580)
```

```
# created checkbutton
```

```
self.chb = Checkbutton(self.window7,  
text="Agree",bg="gold", command=self.Msg) #  
function for message  
self.chb.place(x=700, y=580)  
  
# funtion for message box  
def Msg(self):  
    messagebox.showinfo("Details", f"Agreed the  
Terms & Condition and Privacy Policy")  
window = Tk()    # created object for Tk  
obj = Cab(window) # created object for class Cab  
window.mainloop() # Displaying the GUI on to the  
console
```

5. Conclusion

The end product is obtained that includes all the mentioned modules discussed earlier.

Learnt to make GUI using Tkinter in python.

Learnt to implement database connectivity using sqlite3.

The project is capable of taking booking of cab and show the details field .by user.

6. Bibliography

<https://www.geeksforgeeks.org/sql-using-python/>

<https://www.w3schools.com/python/>

<https://stackoverflow.com/>

<https://tutorialspoint.com/>

www.quora.com/

www.reddit.com/

www.google.co.in/

<https://javatpoint.com/>

