**A PROJECT REPORT**

**ON**

**CAB BOOKING SYSTEM**

**SUBMITTED TO:**

**Ranjith Kaur**



**School of Computer Science and Engineering**

Lovely Professional University

Phagwara, Punjab (India)

Submitted By:

A.C.N.V.S.Akhil (K17DB-18)

**CONTENTS**

**Topic Page No**

* Introduction 1-2
* Modules 3
* Source Code 4 – 12
* Screenshots of Project 13 – 16
* Conclusion 17

**INTRODUCTION**

This project is to make a python based cab booking system for LPU campus only. The project will consist of few modules which will provide the features of booking, cancelling the cab along with the payment option. This project will provide the facility to book cab from anywhere in campus. This project car booking system project is to maintain records of cars. Basically this system will help car booking shopper to make daily record and easy billing of customers and also help to keep maintain monthly revenues and help to grow business. This system works 24×7.

**MODULES**

There are 3 basic Modules:

* Login – This module are for the regular customers. They login and get the required services.
* New User – This module is for the new users, who are new to the software, this module fetches the details and then creates account validation and verification.
* Available Routes – This is the module which shows the available routes for the required travel.
* Administrator Module
* Administrator Entry
* Register Module

**SOURCE CODE**

**Administrator**

import tkinter

from tkinter import\*

import sqlite3

import os

from tkinter import messagebox

interface=Tk()

interface.title("ADMIN")

interface.geometry("300x300")

username=StringVar()

password=StringVar()

def back():

interface.destroy()

os.system('login.py')

def submit():

j=0

try:

conn=sqlite3.connect('admin\_data.db')

print("CONNECTED")

except:

print("NOT CONNECTED")

m=conn.execute("select Username,Password from data")

user=username.get()

key=password.get()

if(len(user)==0 or len(key)==0):

messagebox.showinfo("!!ERROR!!", "PLEASE FILL IN THE DATA")

obj.func\_1()

else:

pass

if(len(key)<6):

messagebox.showinfo("!!ERROR!!","WRONG PASSWORD")

obj.func\_1()

else:

pass

for i in m:

if(user==i[0]):

if(key==i[1]):

conn.commit()

interface.destroy()

os.system('routes.py')

else:

messagebox.showinfo("!!ERROR!!","PASSWORD IS INCORRECT")

else:

messagebox.showinfo("!!ERROR!!","USERNAME NOT PRESENT")

Label(interface,text="USERNAME: ",font='cambria').place(x=15,y=100)

Entry(interface,textvar=username,font='cambria',bd=2).place(x=105,y=100)

Label(interface,text="PASSWORD: ",font='cambria').place(x=15,y=150)

Entry(interface,textvar=password,font='cambria',bd=2,show="\*").place(x=105,y=150)

class A:

def func\_1(self):

Button(interface,text="SUBMIT",command=submit).place(x=130,y=210)

Button(interface,text="<--BACK",command=back).place(x=200,y=210)

obj=A()

obj.func\_1()

interface.mainloop()

**Login**

import tkinter

from tkinter import\*

import sqlite3

import os

from tkinter import messagebox

interface=Tk()

interface.title("LOGIN")

interface.geometry("300x300")

email=StringVar()

password=StringVar()

def register():

interface.destroy()

os.system('new\_user.py')

def admin():

interface.destroy()

os.system('admin.py')

def submit():

j=0

try:

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

print("CONNECTED")

except:

print("NOT CONNECTED")

m=conn.execute("select Mail\_Address,Password from data")

mail=email.get()

key=password.get()

if(len(mail)==0 or len(key)==0):

messagebox.showinfo("!!ERROR!!", "PLEASE FILL IN THE DATA")

obj.func\_1()

if ('@' not in mail):

messagebox.showinfo("!!ERROR!!", "PLEASE PROVIDE A VALID MAIL ID")

obj.func\_1()

else:

pass

if(len(key)<6):

messagebox.showinfo("!!ERROR!!","PLEASE ENSURE LENGTH OF PASSWORD(ATLEAST 6)")

obj.func\_1()

else:

pass

for i in m:

if(mail==i[0]):

if(key==i[1]):

conn.commit()

interface.destroy()

os.system('booking.py')

else:

messagebox.showinfo("!!ERROR!!","PASSWORD IS INCORRECT")

else:

messagebox.showinfo("!!ERROR!!","MAIL ID IS INCORRECT")

Label(interface,text="EMAIL: ",font='cambria').place(x=10,y=100)

Entry(interface,textvar=email,font='cambria',bd=2).place(x=105,y=100)

Label(interface,text="PASSWORD: ",font='cambria').place(x=10,y=150)

Entry(interface,textvar=password,font='cambria',bd=2,show="\*").place(x=105,y=150)

class A:

def func\_1(self):

Button(interface,text="LOGIN",command=submit).place(x=150,y=210)

Button(interface,text="REGISTER",command=register).place(x=210,y=210)

Button(interface,text="ADMIN",command=admin).place(x=80,y=210)

obj=A()

obj.func\_1()

interface.mainloop()

**Register**

import tkinter

from tkinter import\*

import sqlite3

import os

from tkinter import messagebox

interface=Tk()

interface.title("REGISTER")

interface.geometry("675x600")

first\_name=StringVar()

second\_name=StringVar()

gender=IntVar()

number=IntVar()

mail=StringVar()

password1=StringVar()

password2=StringVar()

def login():

interface.destroy()

os.system('login.py')

def submit():

j=0

name\_1=first\_name.get()

name\_2=second\_name.get()

gen=gender.get()

try:

phone=number.get()

except:

messagebox.showinfo("!!ERROR!!","PLEASE ENTER A VALID phone number")

obj.func\_1()

email=mail.get()

password=password1.get()

if len(first\_name.get()) == 0 or len(second\_name.get()) == 0 or gender.get() == 0 or len(mail.get()) == 0 or len(password1.get()) == 0 or len(password2.get()) == 0 or password2.get()!= password1.get():

messagebox.showinfo("!!ERROR!!", "PLEASE FILL IN THE DATA OR CHECK THE PASSWORD")

obj.func\_1()

else:

pass

if ('@' not in email):

messagebox.showinfo("!!ERROR!!", "PLEASE PROVIDE A VALID MAIL ID")

obj.func\_1()

else:

pass

if(len(password)<6):

messagebox.showinfo("!!ERROR!!","PLEASE ENSURE LENGTH OF PASSWORD(ATLEAST 6)")

obj.func\_1()

else:

if(gen == 1):

gen='MALE'

else:

gen='FEMALE'

try:

conn = sqlite3.connect("database.db")

print("DATABASE CONNECTED")

except Error as e:

print(e)

#conn.execute("create table data(First\_Name varchar(50),Second\_Name varchar(50),Gender varchar(6),Phone\_Number number(10),Mail\_Address varchar(50),Password varchar(50));")

conn.execute("insert into data(First\_Name,Second\_Name,Gender,Phone\_Number,Mail\_Address,Password)values(?,?,?,?,?,?)",(name\_1,name\_2,gen,phone,email,password))

conn.commit()

interface.destroy()

os.system('login.py')

Label(interface,text="FIRST NAME",font='cambria').place(x=150,y=100)

Entry(interface,textvar=first\_name,font='cambria',bd=2).place(x=350,y=100)

Label(interface,text="SECOND NAME",font='cambria').place(x=150,y=150)

Entry(interface,textvar=second\_name,font='cambria',bd=2).place(x=350,y=150)

Radiobutton(interface, text="MALE",padx = 20, variable=gender,value=1,font='cambria').place(x=150,y=200)

Radiobutton(interface, text="FEMALE",padx = 20, variable=gender, value=2,font='cambria').place(x=350,y=200)

Label(interface,text="MOBILE NO",font='cambria').place(x=150,y=250)

Entry(interface,textvar=number,font='cambria',bd=2).place(x=350,y=250)

Label(interface,text="MAIL ID",font='cambria').place(x=150,y=300)

Entry(interface,textvar=mail,font='cambria',bd=2).place(x=350,y=300)

Label(interface,text="PASSWORD",font='cambria').place(x=150,y=350)

Entry(interface,textvar=password1,font='cambria',bd=2,show="\*").place(x=350,y=350)

Label(interface,text="CONFIRM PASSWORD",font='cambria').place(x=150,y=400)

Entry(interface,textvar=password2,font='cambria',bd=2,show="\*").place(x=350,y=400)

class A:

def func\_1(self):

Button(interface,text="SUBMIT",command=submit).place(x=390,y=450)

obj=A()

obj.func\_1()

Button(interface,text="LOGIN",command=login).place(x=450,y=450)

interface.mainloop()

**Routes**

import tkinter

from tkinter import\*

import sqlite3

import os

from tkinter import messagebox

interface=Tk()

interface.title("ROUTES")

interface.geometry("600x600")

location=StringVar()

destination=StringVar()

cost=IntVar()

time=IntVar()

distance=IntVar()

def submit():

j=0

try:

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

print("CONNECTED")

except:

print("NOT CONNECTED")

m=conn.execute("select Location,Destination from data")

loc=location.get()

des=destination.get()

try:

cost\_1=cost.get()

time\_1=time.get()

path=distance.get()

except:

messagebox.showinfo("!!ERROR!!", "PLEASE ENTER CORRECT COST")

obj.func\_1()

if(len(loc)==0 or len(des)==0 or cost\_1 ==0 or time==0 or distance==0):

messagebox.showinfo("!!ERROR!!", "PLEASE FILL IN THE DATA")

obj.func\_1()

else:

for i in m:

if(loc==i[0] and des ==i[1]):

messagebox.showinfo("!!ERROR!!", "PLEASE FILL NEW DATA")

obj.func\_1()

else:

pass

# conn.execute("create table data(Location varchar(50),Destination varchar(50), Fare number(10), Time number(10),Distance number(10));")

conn.execute("insert into data(Location,Destination,Fare,Time,Distance) values(?,?,?,?,?)",(loc,des,cost\_1,time\_1,path))

conn.execute("insert into data(Location,Destination,Fare,Time,Distance) values(?,?,?,?,?)",(des,loc,cost\_1,time\_1,path))

conn.commit()

interface.destroy()

os.system('available\_routes.py')

Label(interface,text="LOCATION: ",font='cambria').place(x=15,y=100)

Entry(interface,textvar=location,font='cambria',bd=2).place(x=105,y=100)

Label(interface,text="DESTINATION: ",font='cambria').place(x=15,y=150)

Entry(interface,textvar=destination,font='cambria',bd=2).place(x=105,y=150)

Label(interface,text="COST: ",font='cambria').place(x=15,y=200)

Entry(interface,textvar=cost,font='cambria',bd=2).place(x=105,y=200)

Label(interface,text="TIME: ",font='cambria').place(x=15,y=250)

Entry(interface,textvar=time,font='cambria',bd=2).place(x=105,y=250)

Label(interface,text="DISTANCE: ",font='cambria').place(x=15,y=300)

Entry(interface,textvar=distance,font='cambria',bd=2).place(x=105,y=300)

class A:

def func\_1(self):

Button(interface,text="SUBMIT",command=submit).place(x=200,y=350)

obj=A()

obj.func\_1()

interface.mainloop()

**Available Routes**

import tkinter

from tkinter import\*

import sqlite3

import os

from tkinter import messagebox

interface=Tk()

interface.title("AVAILABLE ROUTES")

interface.geometry("1000x600")

conn=sqlite3.connect("routes.db")

m=conn.execute("select Location,Destination,Fare,Time,Distance from data")

j=1

def exit():

interface.destroy()

def login():

interface.destroy()

os.system('login.py')

Label(interface,text=" LOCATION DESTINATION COST TIME DISTANCE",font='cambria',bd=2).pack(side='top')

for i in m:

a=["LOCATION : ",i[0],"DESTINATION : ",i[1],"COST : ",i[2],"TIME : ",i[3],"DISTANCE :",i[4]]

Label(interface,text=a,font='cambria',bd=2).pack()

conn.commit()

Button(interface,text="EXIT",command=exit).pack(side='bottom', padx = 5, pady = 5);

Button(interface,text="<--BACK",command=login).pack(side='bottom', padx = 5, pady = 5)

interface.mainloop()

**Booking**

import tkinter

from tkinter import\*

import sqlite3

import os

from tkinter import messagebox

interface=Tk()

interface.title("LOGIN")

interface.geometry("300x300")

time=StringVar()

date=StringVar()

location=StringVar(interface)

des\_data=tuple()

destination=StringVar(interface)

def ok():

des\_data=location.get()

print(des\_data)

print(type(loc[0]))

print(type(des\_data))

conn=sqlite3.connect("routes.db")

c = conn.cursor()

m=("select Location from data")

c.execute(m)

loc=list(c.fetchall())

location.set(loc[0])

OptionMenu(interface,location,\*loc).place(x=100,y=100)

Button(interface, text="OK", command=ok).place(x=240,y=100)

n=("select Destination from data where Location=?",(loc[0]))

c.execute(n)

des=list(c.fetchall())

destination.set(des[0])

print(des[0])

OptionMenu(interface,destination,\*des).place(x=100,y=180)

conn.commit()

Button(interface,text="EXIT",command=exit).pack(side='bottom', padx = 5, pady = 5);

interface.mainloop()

**Main**

import tkinter

from tkinter import\*

import sqlite3

import os

interface=Tk()

interface.title('LPU CAB BOOKING SYSTEM')

interface.geometry("1350x750")

#function 1

def login():

interface.destroy()

os.system('login.py')

#function 2

def new\_user():

interface.destroy()

os.system('new\_user.py')

#function 3

def available\_routes():

interface.destroy()

os.system('available\_routes.py')

#buttons in interface

Button(interface,text="LOGIN",command=login,bg="red", fg="white").place(x=630,y=200)

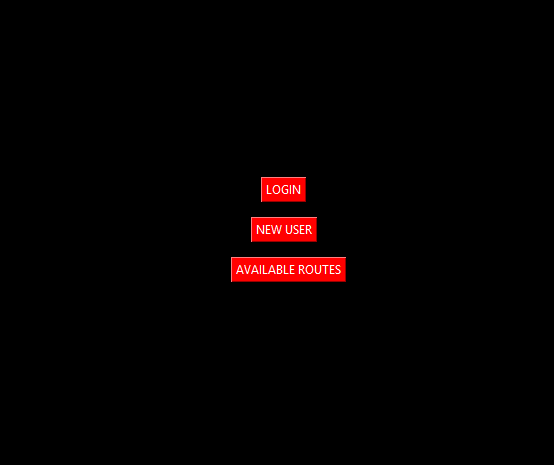
Button(interface,text="NEW USER",command=new\_user,bg="red", fg="white").place(x=620,y=240)

Button(interface,text="AVAILABLE ROUTES",command=available\_routes,bg="red", fg="white").place(x=600,y=280)

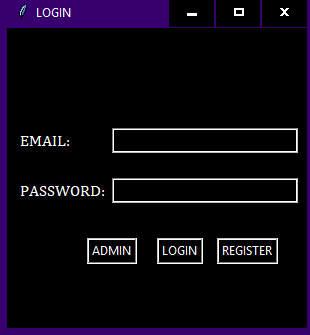
interface.mainloop()

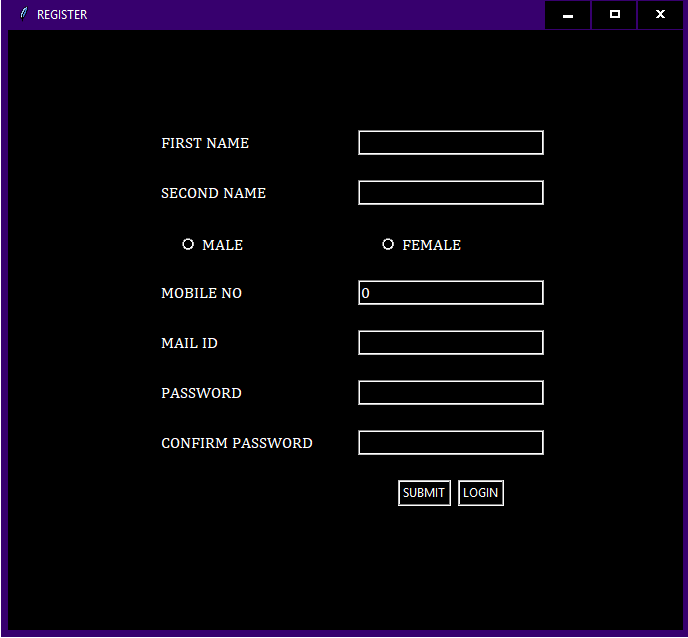
**SCREENSHOTS OF CAB BOOKING SYSTEM**

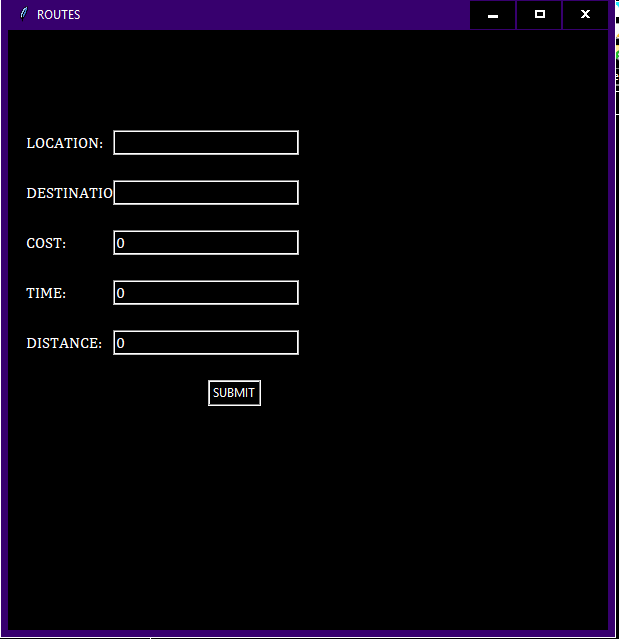
**Main**

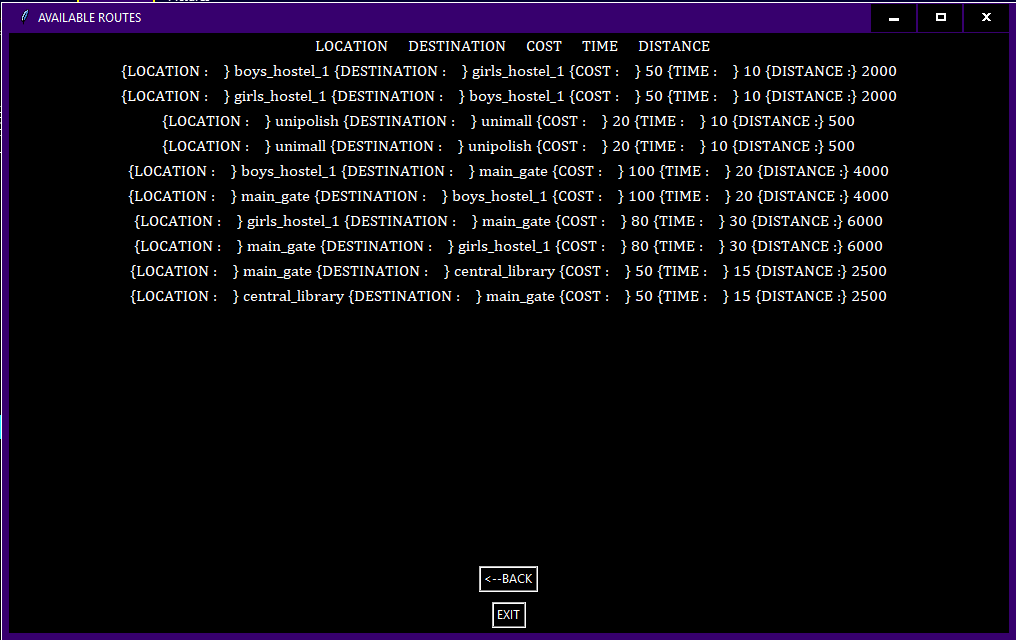
****

**Login**

****

**Register**

**Routes**

**Available Routes**

**CONCLUSION**

In this project we have made an application based on python language, specifically on python 3.6.3 64 bit.

Our project provides basic 3 three interface for new user, existing user and the admin and this main interfaces are further divided which contains the booking option, details regarding the routes.

Finally, we are heading towards making an application which will be used for cab booking.

* Cab Booking System is an application and it is restricted to only limited type of users.
* In this application different types of Users have been given access rights and they are restricted up to their functionalities
* Here The Details of Users and cabs are maintained.