**PROJECT REPORT**

**On**

**Cab booking in LPU**

**BACHELOR OF TECHNOLOGY**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

By

|  |  |
| --- | --- |
| **Name** | **Roll No.:** |
| **Cheela Shiva Sai** | **43** |
| **Gade Vinod** | **44** |
| **Ch.Vishnuvardhan reddy** | **45** |



**School of Computer Science and Engineering**

Lovely Professional University

Phagwara, Punjab (India)

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Serial No.** | **Chapters** | **Page Number** |
|  | **ACKNOWLEDGEMENT** | **3** |
|  | **INTRODUCTION** | **4** |
|  | **OBJECTIVE** | **8** |
|  | **SCREENSHOTS OF GUI** | **10** |
|  | **SOURCE CODE** | **17** |
|  | **RESULT** | **40** |
|  | **REFERENCES** | **41** |

**ACKNOWLEDMENT**

The success and final outcome of this project required a lot of guidance and assistance from many people .We are extremely privileged to have got this all along the completion of my project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We respect and thank **Ms. Gagandeep Kaur** for providing us an opportunity to do the project work on **System for booking Cabs** and giving us all support and guidance which made me complete the project duty. We are extremely thankful to her for providing such a nice support and guidance.

Secondly we would also like to thank our parents and friends who helped us a lot in finalizing this project within the limited time frame

**INTRODUCTION:**

Cab Booking management System is developed to manage all cab hiring work. It useful for cab booking agency that are specialized in Hiring cabs to customers. Using this system many cab-booking agency are moving ahead to become a pioneer in the car rental industry by completely focusing on customers. Using this system it is very easy for customer to book a cab and cab-booking agency can also track their booking. So it is also very useful for cab booking agency. It is a system through which customers can view available cabs; register the cabs, view profile and book cabs. Mostly people use cab service for their daily transportations need. Cab booking agency can also check which car is free for booking and which cars are on booking at present time. The objective and scope of my project cab booking System is to record the details various activities of user. It will simplify the task and reduce the paper work. Using this car booking management system car owner can also become partner of car booking agency by giving their car for booking. From this system car rental company can manage all car bookings and customer information. User can book cars and admin can confirm the booking and cancel the booking on the basis of availability of the cars and drivers.

We have develop this system to produce a system that allow customer to register and reserve cab and for the company to effectively manage their Cab hiring business. Presently car booking agency do all work offline when a customer comes to them they take the booking order and call the car driver to check their availability with their car if they manage to find a car for booking they confirm the order otherwise they cancel the order as they have no car for the booking. This process waste a lot of time of customer and also of car booking agency and it also give bad name to the agency but with our system car agency can confirm the order within a minute by checking the availability of cars for booking. So this cab booking system is helpful to ease customer’s task whenever they need to rent a cab or hire a cab.

**How cab booking management system work?**

When user want to hire a car he contact the car rental service. He can select date of journey .To book a car user have to enter his mobile no. and other ID information. This system keeps all the details of cab booking and when customer booking comes then he check the system for available car and confirm the booking if car is available on the booking date. User have to give information such as dates of journey, mobile number. User also has to give pick up point information and drop off information to book a car. This system make so easy to manage cab booking for the cab booking agency that ageny can focus on service quality like how to provides service with well-conditioned new vehicles, with experience drivers for a happy journey of the customers. This project intends to introduce more user friendly in the various activities such as record updating, maintenance, and searching.

Cab booking system also provide services to corporate houses. This system also takes records of the car and their permits and other documents. All cabs have proper permits and documentation so that the clients could not be hassled for the lack of documents. This system also keep records of the car drivers and admin can check driver related information like his name, id documents and which car he is driving on booking etc. Cab drivers are educated, polite, and reliable and are trained to handle acute breakdowns. The cab service includes all categories of cars from luxury to budget.

cab rental system also helpful to maintain the quality of service by tracking the records of user’s feedback of the service.

**Proposed cab booking management system**

Functionalities and advantages of proposed car booking management system are:

In this Cab booking system all data is centralized which has overcome the Sharing problem in previous system. We provide free registration for the cab owners. And we proved free service to users who want to book a car or cab. As data is maintained online, it is easy for a person to update the details, which has overcome the tedious updating in previous system. Cab booking management is easy and performance is good. Here the traveller can book a cab, taxi and car by viewing all the cab details and pricing details available, according to selected city and area. It is the reliable service provided to both customers and travel agencies. Mainly the system has automated the Transportation Process.

**Online Cab Booking management system has the following services:-**

Users registration: A registration portal to hold customer details, monitor their transaction and used same to offer better and improve services to them. Availability: The software acts as an office that is open 24/7.

**Cab Search-**

Users can search cab for a particular location . To book a car use has to login the system. Users required to enter source, Destination, & place where he wants to go.

**Registration Process-**

User must be registered before booking a cab. Before booking a car user have to enter his contact information like name, address and mobile number. Only a valid user can login the system and book the car by giving his picking up place and drop off place. All the data supply by the user will be stored in database and it will be used for further validations and authenticated. Login names and password will be stored in the database.

OBJECTIVE:

**Book rides according to your plans**

This app helps its users to book their ride according to their day and time.

**Avail rides at their doorstep**

The end users who are looking to book rides can avail a ride with doorstep pickup and drop at an exact location with easy set up a pickup and drop location.

**No bargain on prices**

This app has the feature of fare estimator to let know its user the exact price for a ride for, before confirmation for booking. And once the ride is successfully completed, it will raise the bill for the exact amount to be paid to the driver on the app.

**Pay on a convenient way**

This app will have multiple payment options for its end users to make their fare payments. Payments can be made through cash, credit, wallet and payment gateway.

**Be informed about all actions on your ride booking**

This app will notify your users on confirmed ride bookings and notifies that particular driver. Also lets that particular end user to know their status of booking with instant notifications.

**Know your ride history and avail secure rides**

This app will have the feature of ride history to let its end users know about their previous availed rides. This app will also provides secured rides for its end user with the feature of OTP verification with a driver.

**Earn credits from invites**

This app will enable end users to easily log in with their social account and invites their friends on social network. An end user can earn credit points, which can be used for a discount of price on future rides upon every successful referral.

**Easy submission of documents**

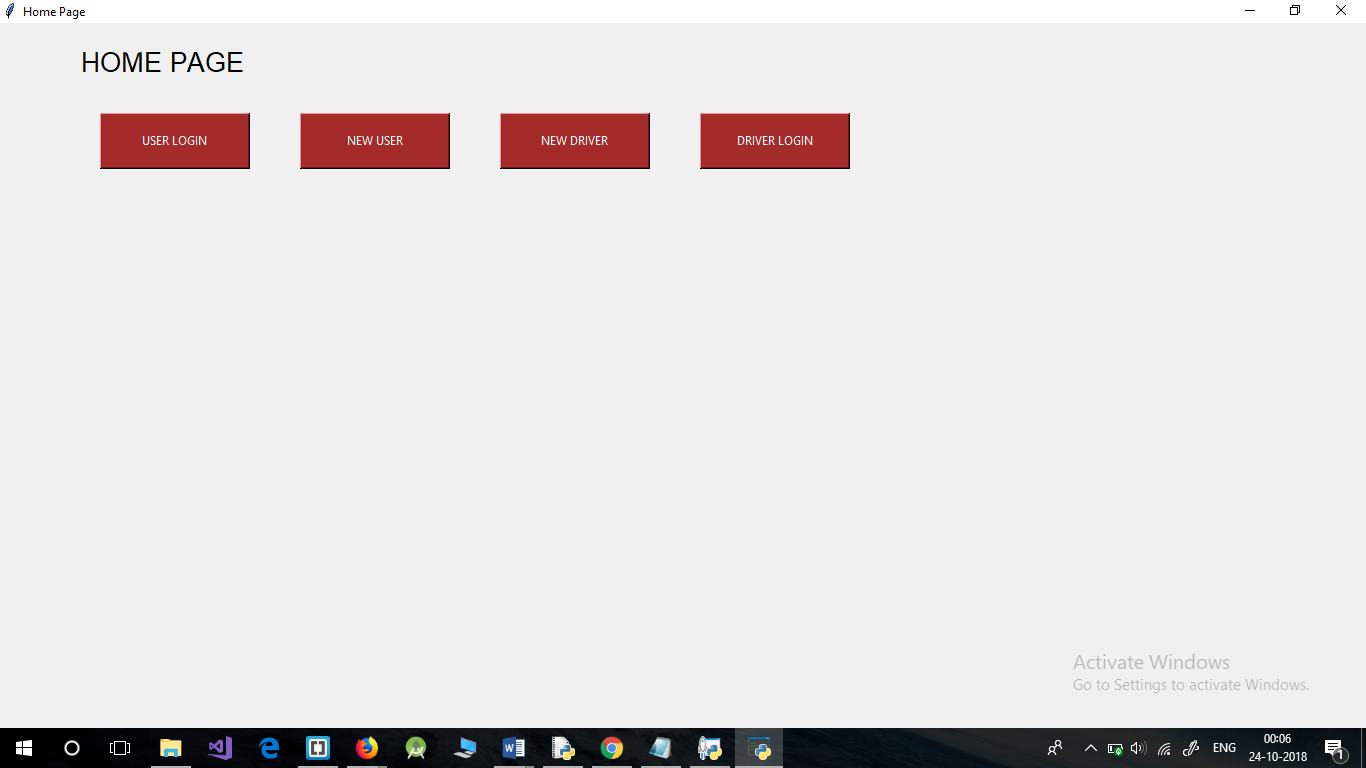
Drivers can easily login to the app and have the document submission feature within the app to easily submit their documents for verification.

**Location experts**

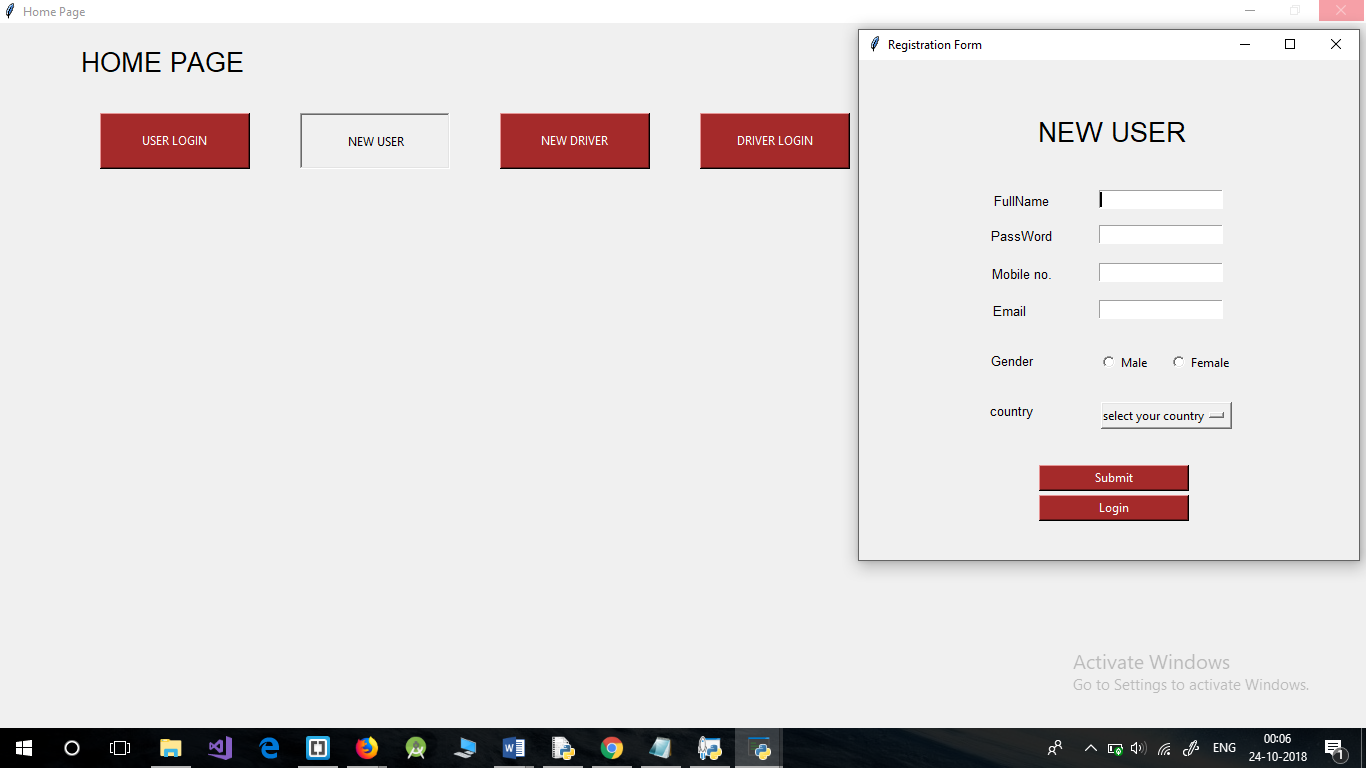
This app can make your drivers a location experts with the feature of live navigation, to help with their ride pick-ups.

ScreenShots:

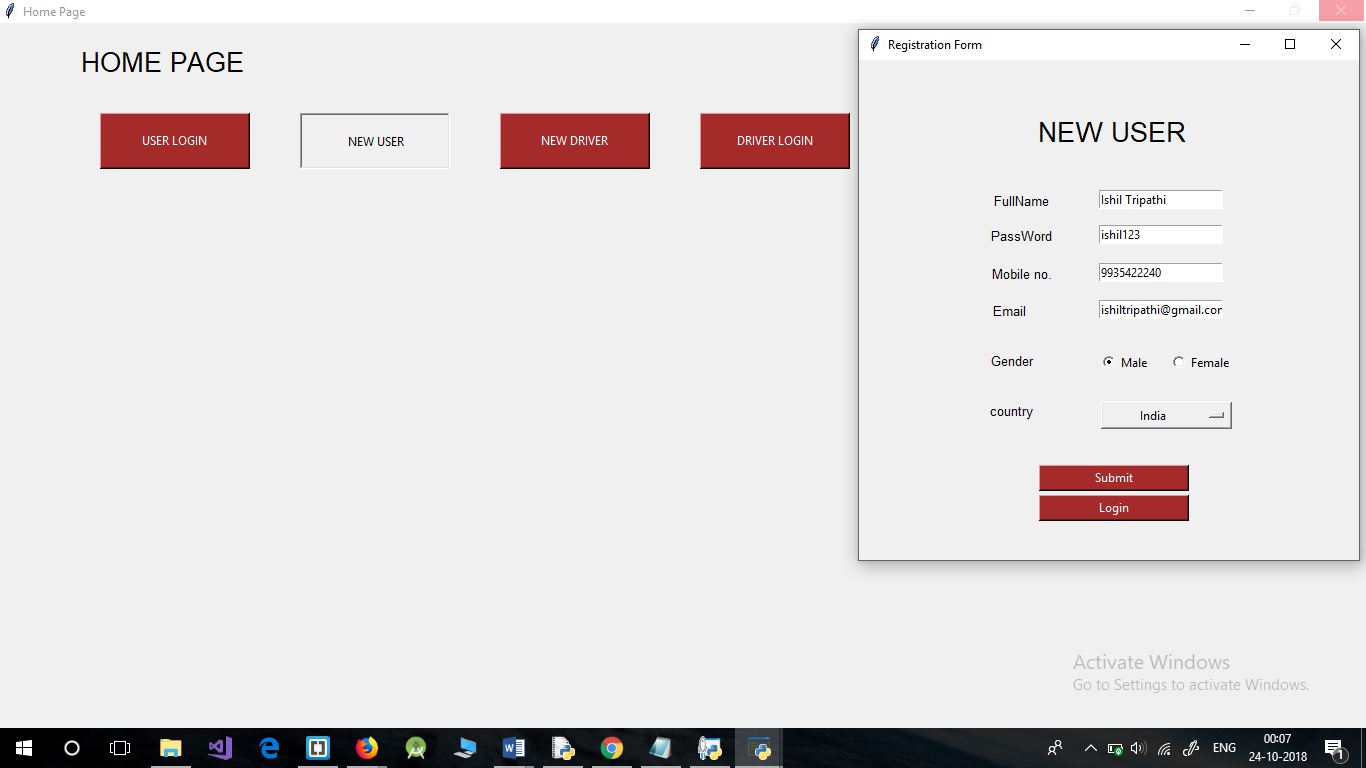
**HOME PAGE:**



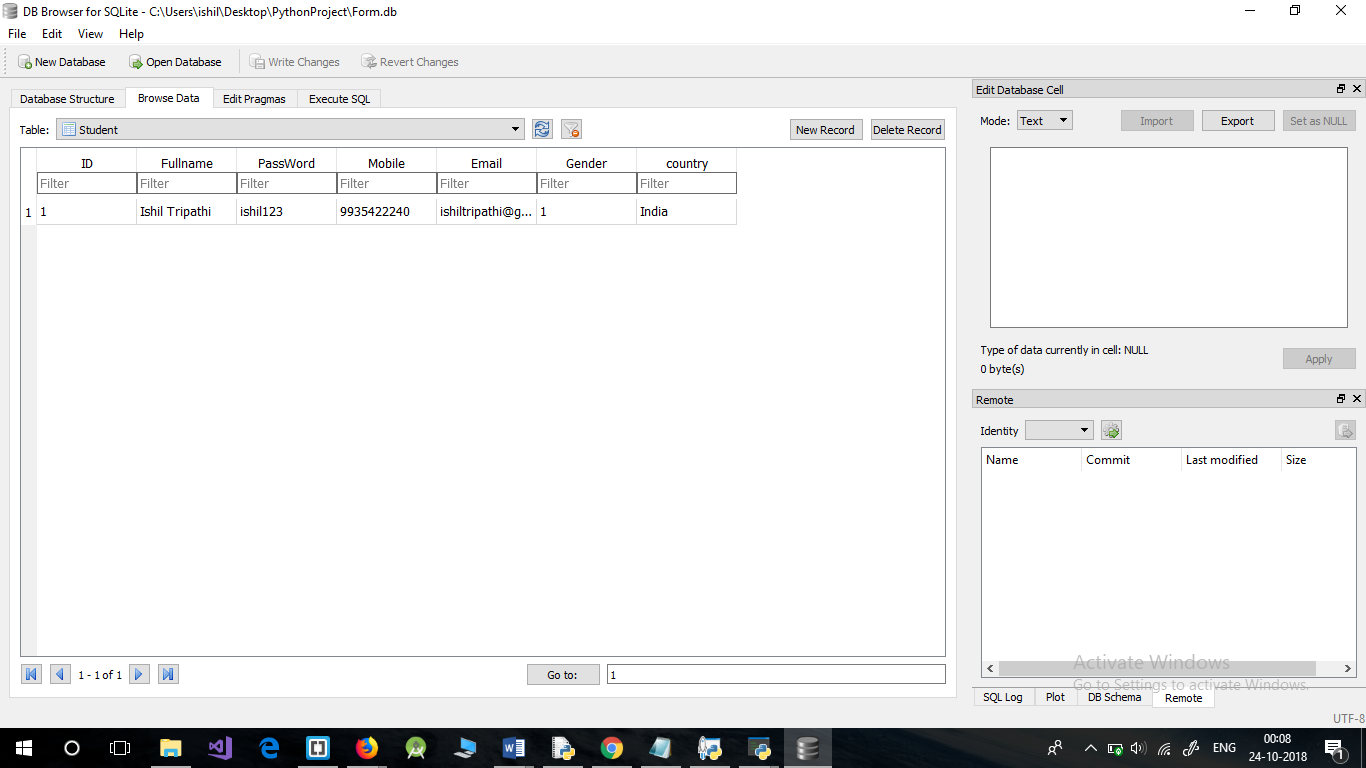
**USER SIGN UP:**



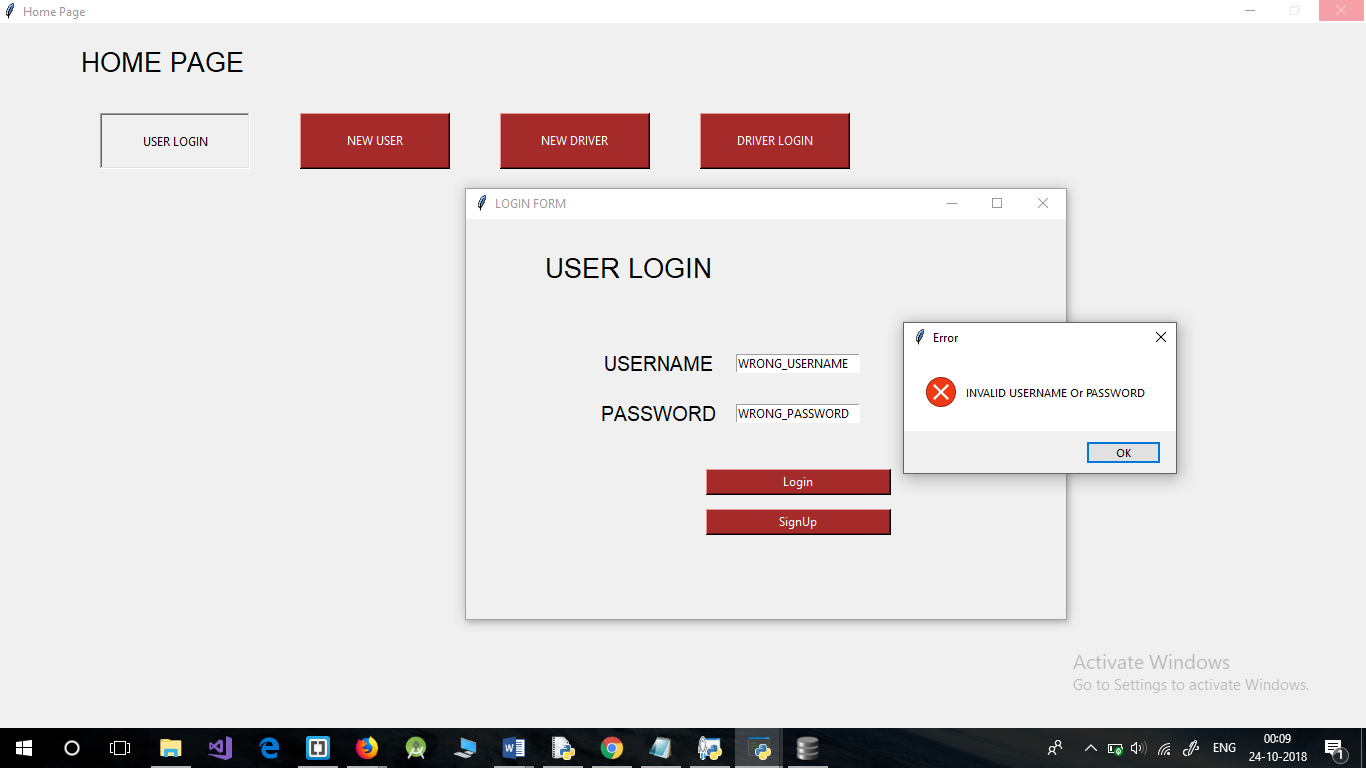
**SIGN\_UP DETAILS FILLED:**



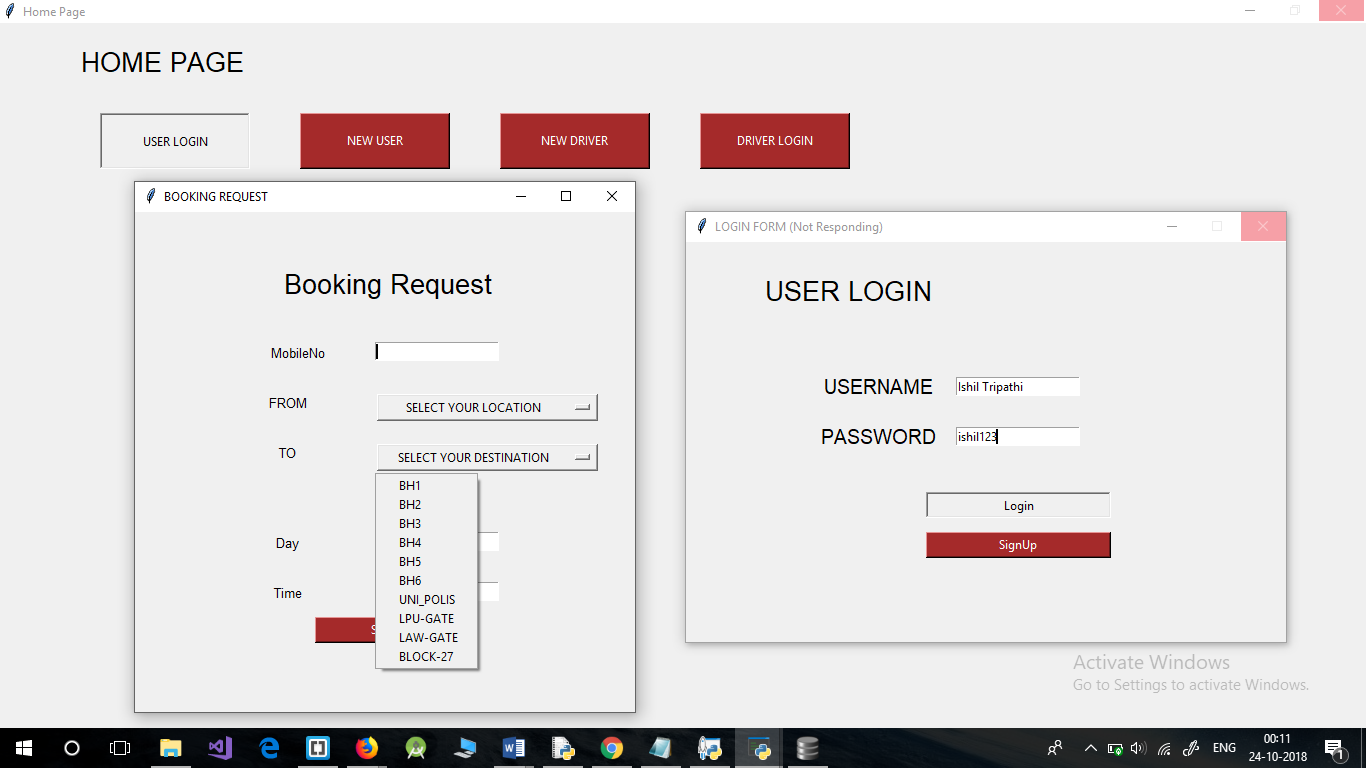
**RECORD UPDATED IN DATABASE:**



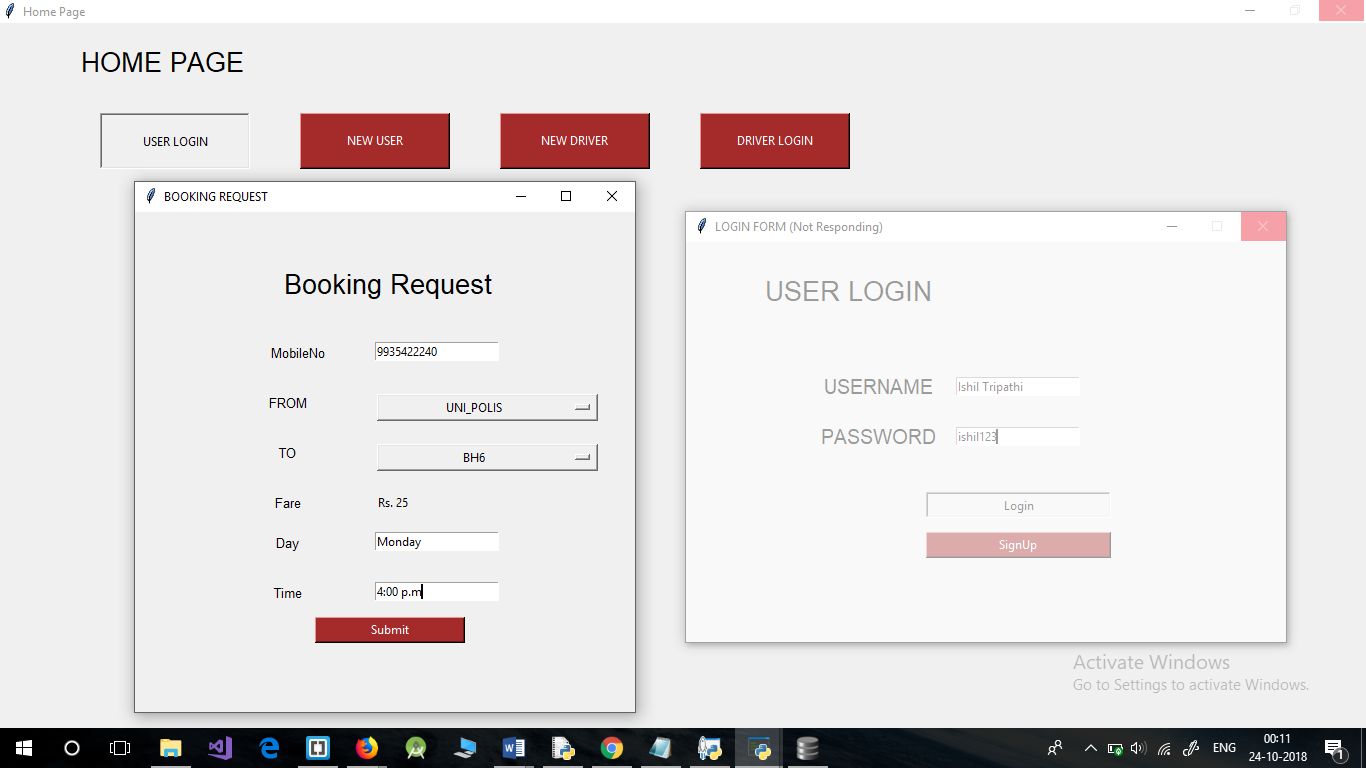
**USER LOGIN WITH INVALID DETAILS:**



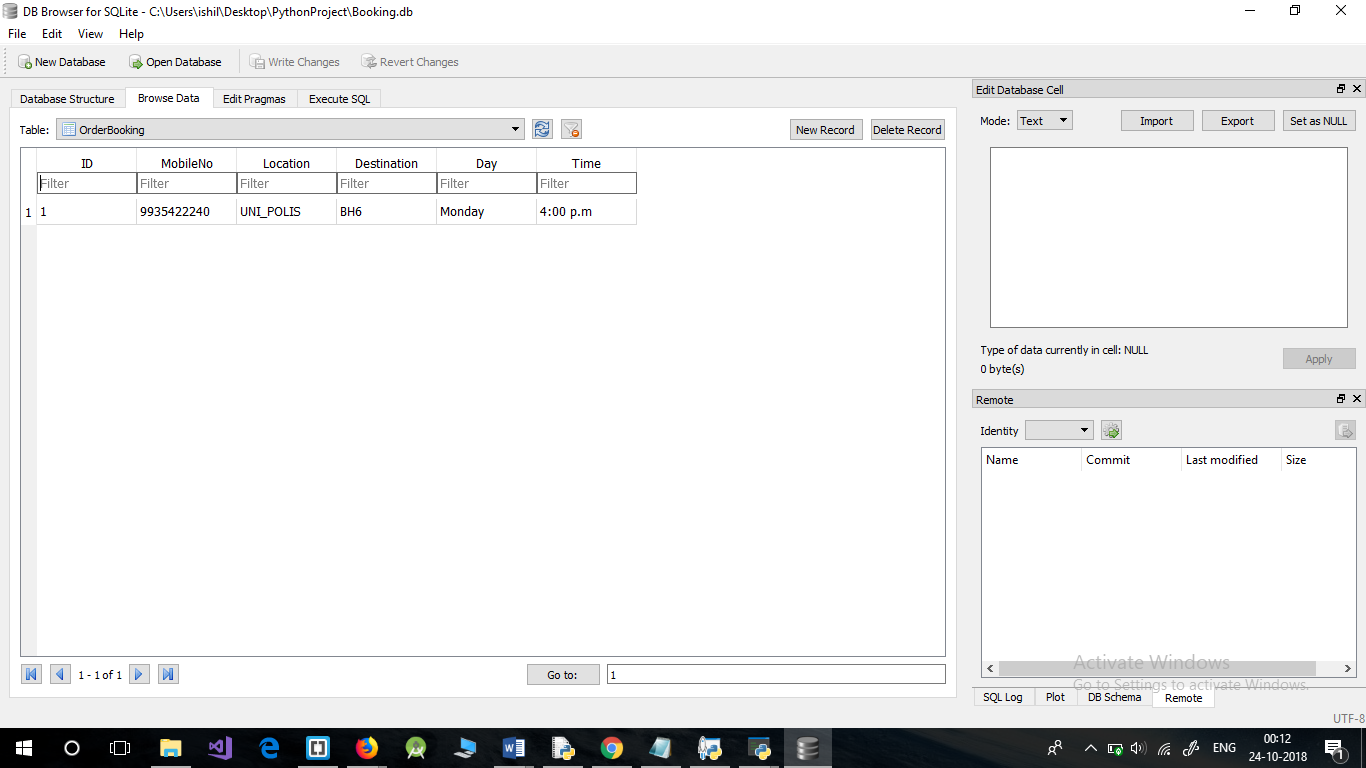
**USER INTERFACE AFTER VALID LOGIN:**



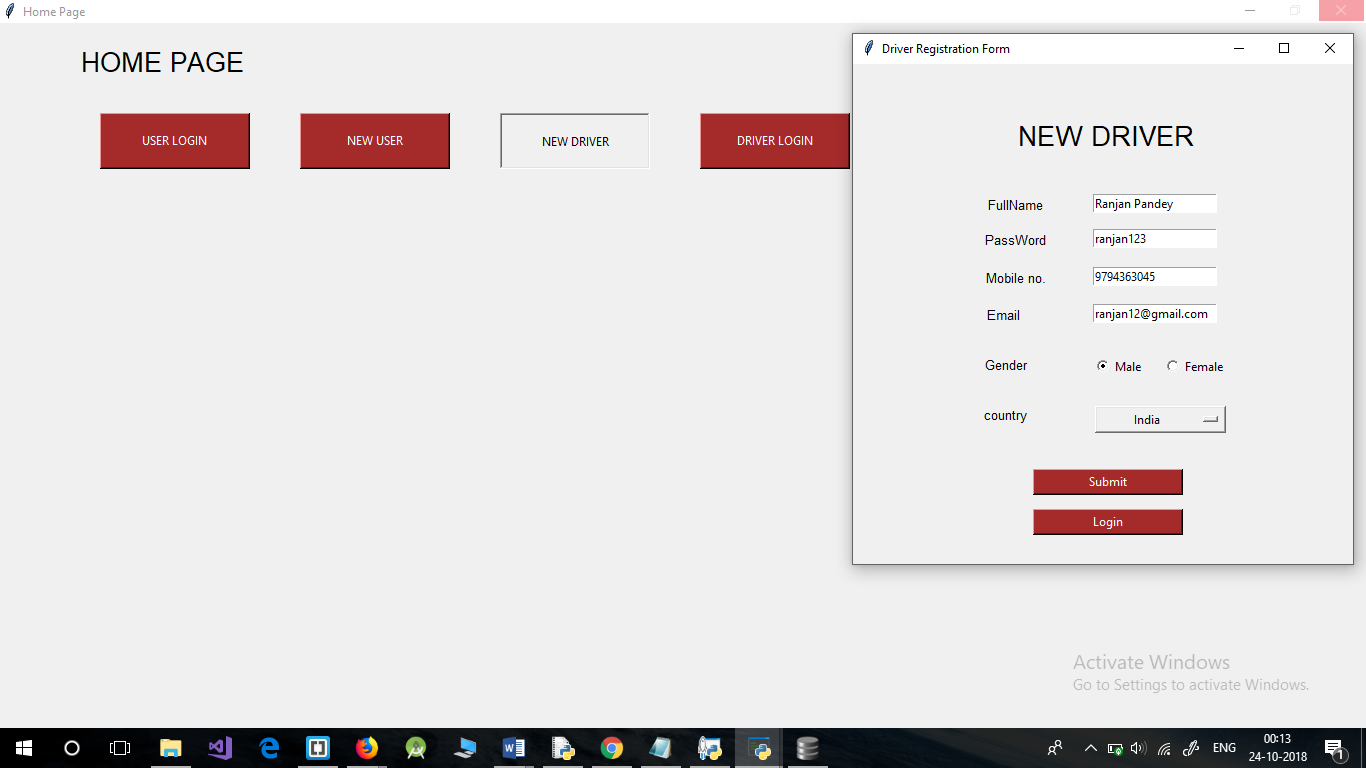
**DETAILS FILLED IN USER INTERFACE:**



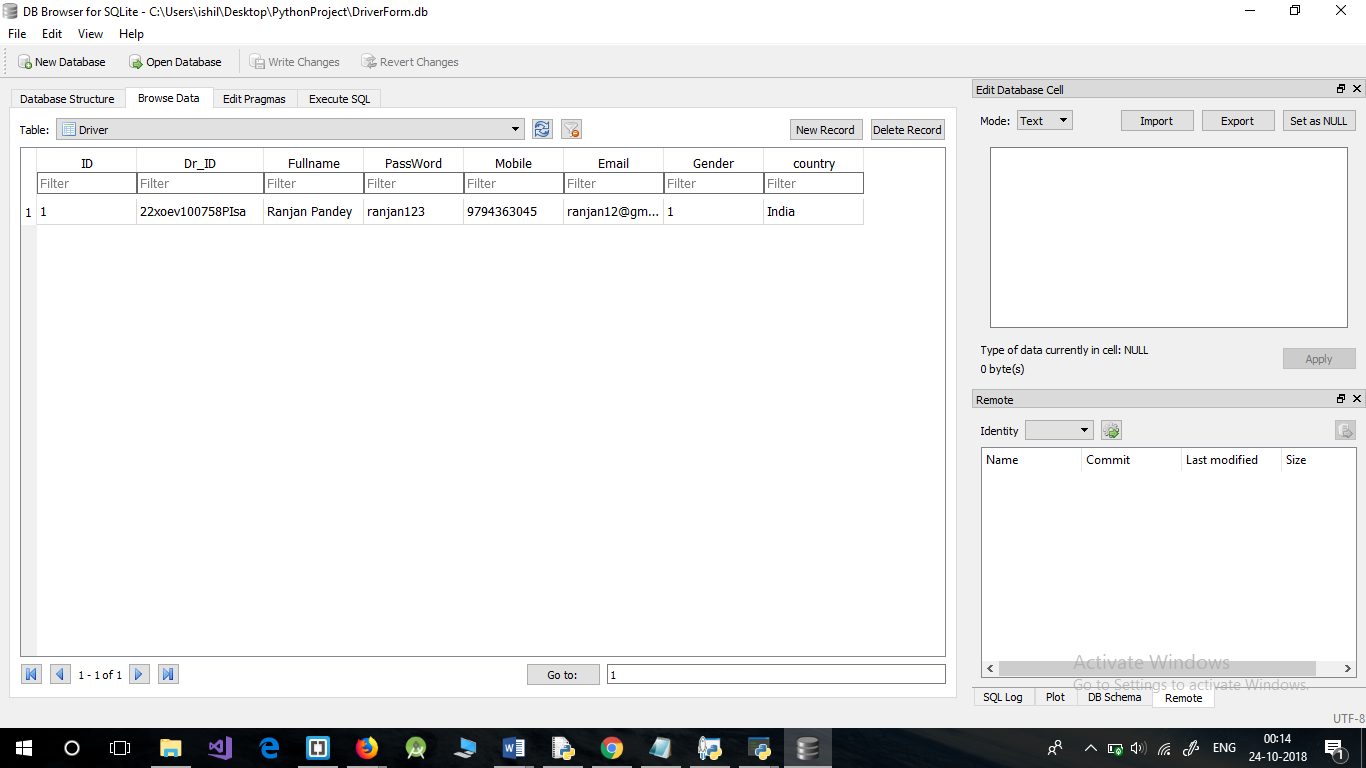
**BOOKING REQUEST UPDATED IN DATABASE:**



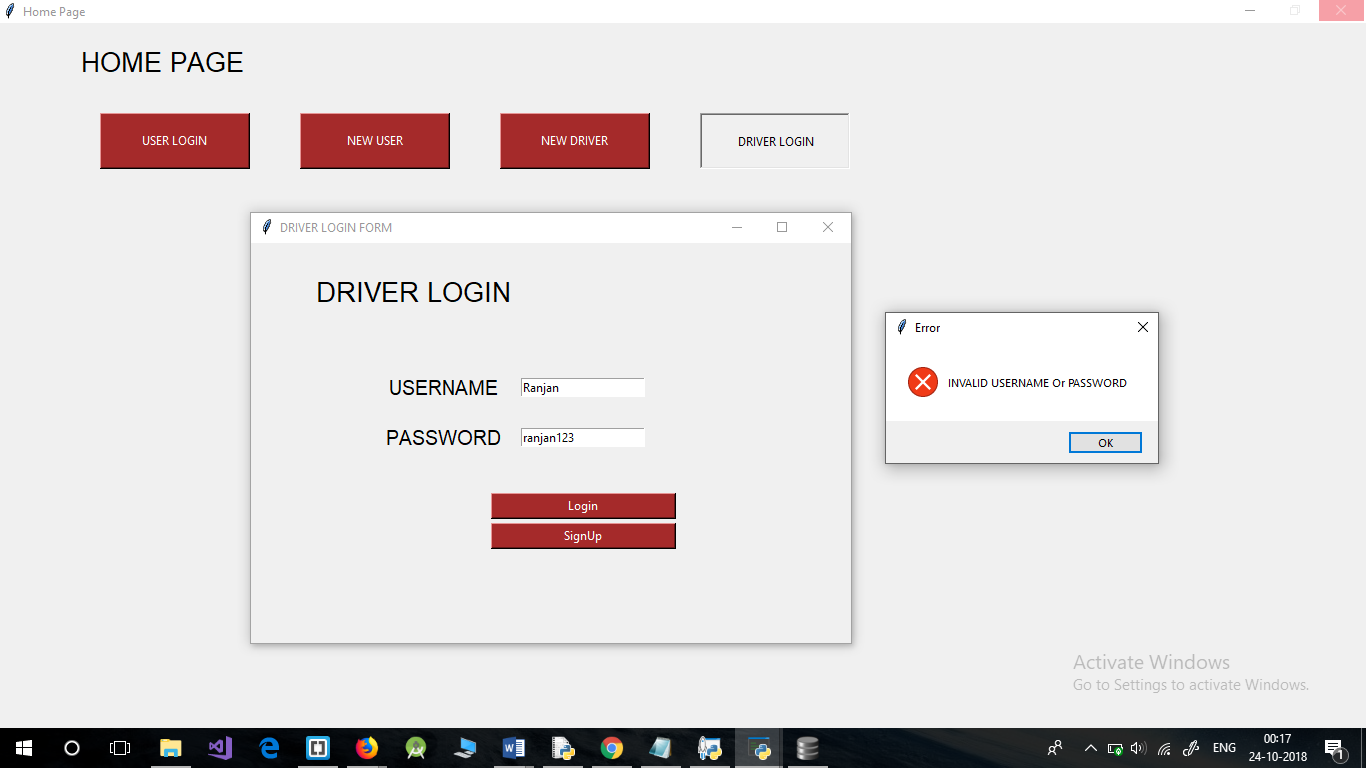
**NEW DRIVER SIGNUP:**



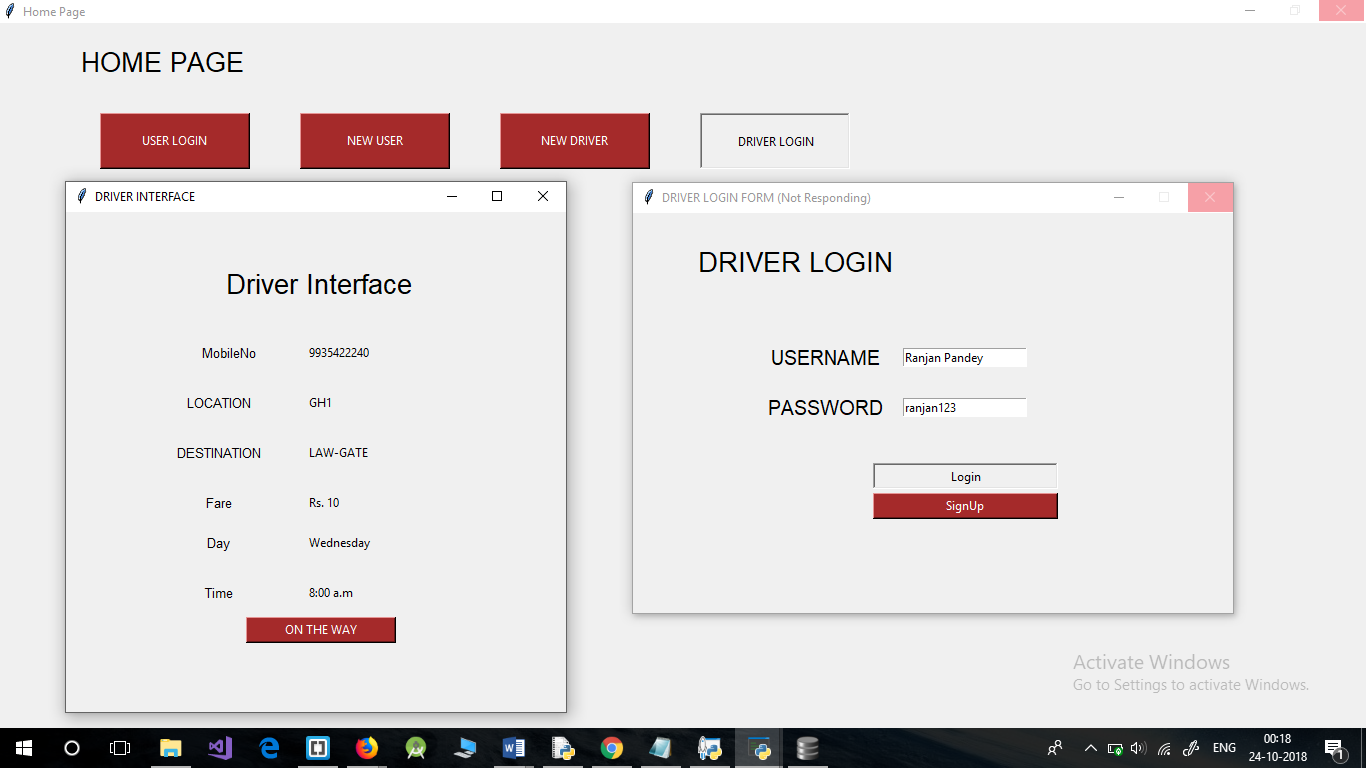
**DRIVER DETAILS ADDED WITH UNIQUE DRID:**



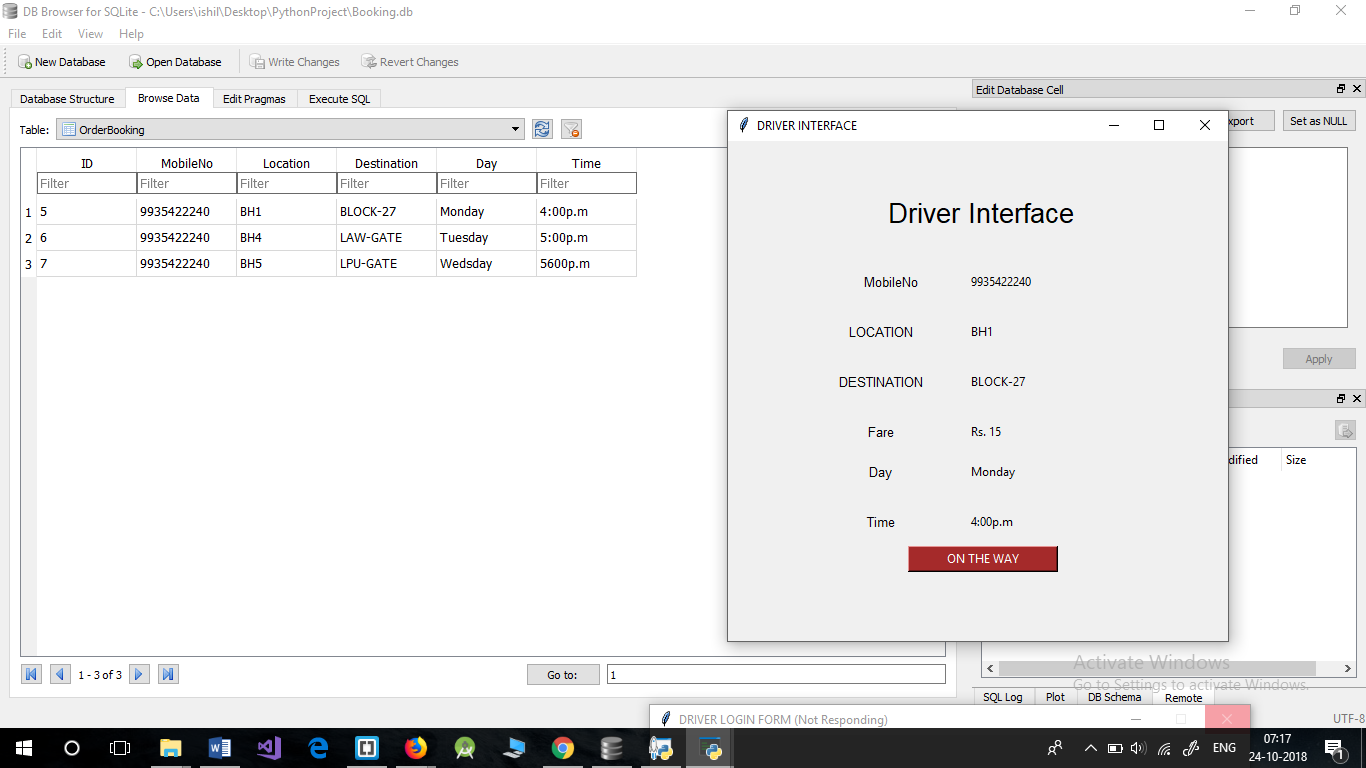
**DRIVER LOGIN:**



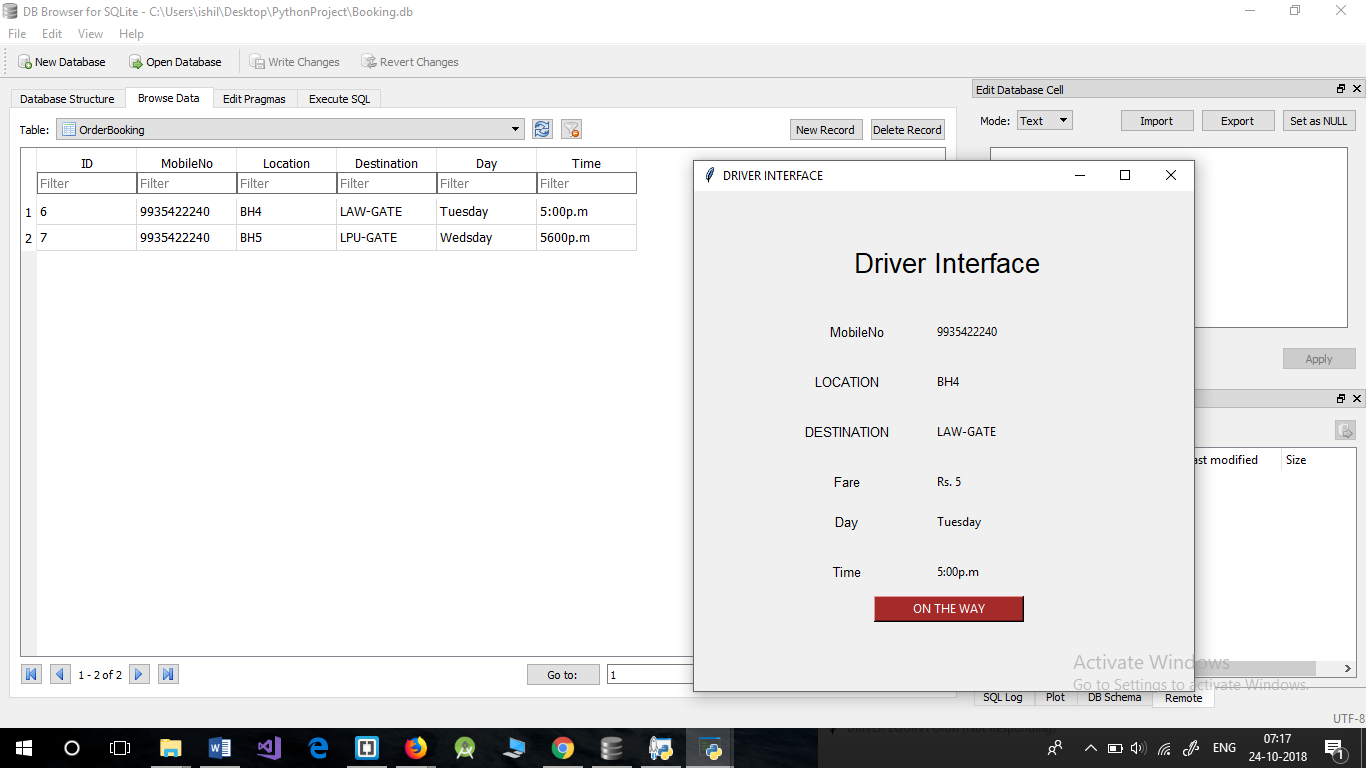
**DRIVER INTERFACE AFTER LOGGED IN:**



**DRIVER INTERFACE AFTER MULTIPLE REQUESTS:**

****

**AFTER HITTING ON THE WAY RECORD WOULD BE DELETED IN QUEUE ORDER:**

****

**CODE:**

**HOME PAGE:**

from tkinter import \*

import os

root = Tk()

root.geometry('1000x200')

root.title("Home Page")

def log():

os.system("LoginUser.py")

def signUp():

os.system("SignUpUser.py")

def driver():

os.system("DriverSignUp.py")

def driver2():

os.system("DriverLogin.py")

label\_0 = Label(root,text="HOME PAGE",width=20,font=("bold",20))

label\_0.place(x=0,y=20)

Button(root,text = "USER LOGIN",width = 20,height = 3,bg='brown',fg='white',command=log).place(x=100,y=90)

Button(root,text = "NEW USER",width = 20,height = 3,bg='brown',fg='white',command=signUp).place(x=300,y=90)

Button(root,text = "NEW DRIVER",width = 20,height = 3,bg='brown',fg='white',command=driver).place(x=500,y=90)

Button(root,text = "DRIVER LOGIN",width = 20,height = 3,bg='brown',fg='white',command=driver2).place(x=700,y=90)

root.mainloop()

**USER SIGN UP:**

from tkinter import \*

import sqlite3

import os

root = Tk()

root.geometry('500x500')

root.title("Registration Form")

Fullname=StringVar()

Email=StringVar()

Password = StringVar()

MobileNo=StringVar()

var = IntVar()

c=StringVar()

def login():

os.system("LoginUser.py")

def database():

name1=Fullname.get()

mob1 = MobileNo.get()

email=Email.get()

gender=var.get()

country=c.get()

password = Password.get()

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

with conn:

cursor=conn.cursor()

cursor.execute('CREATE TABLE IF NOT EXISTS Student (ID INTEGER PRIMARY KEY AUTOINCREMENT,Fullname TEXT UNIQUE,PassWord TEXT UNIQUE,Mobile TEXT,Email TEXT,Gender TEXT,country TEXT)')

cursor.execute('INSERT INTO Student (FullName,PassWord,Mobile,Email,Gender,country) VALUES(?,?,?,?,?,?)',(name1,password,mob1,email,gender,country))

conn.commit()

label\_0 = Label(root, text="NEW USER",width=20,font=("bold", 20))

label\_0.place(x=90,y=53)

label\_1 = Label(root, text="FullName",width=20,font=("bold", 10))

label\_1.place(x=80,y=130)

entry\_1 = Entry(root,textvar=Fullname)

entry\_1.place(x=240,y=130)

label\_pass = Label(root, text="PassWord",width=20,font=("bold", 10))

label\_pass.place(x=80,y=165)

entry\_pass = Entry(root,textvar=Password)

entry\_pass.place(x=240,y=165)

label\_no = Label(root,text = "Mobile no.",width=20,font=("bold",10))

label\_no.place(x=80,y=203)

entry\_no = Entry(root,textvar=MobileNo)

entry\_no.place(x=240,y=203)

label\_2 = Label(root, text="Email",width=20,font=("bold", 10))

label\_2.place(x=68,y=240)

entry\_2 = Entry(root,textvar=Email)

entry\_2.place(x=240,y=240)

label\_3 = Label(root, text="Gender",width=20,font=("bold", 10))

label\_3.place(x=70,y=290)

Radiobutton(root, text="Male",padx = 5, variable=var, value=1).place(x=235,y=290)

Radiobutton(root, text="Female",padx = 20, variable=var, value=2).place(x=290,y=290)

label\_4 = Label(root, text="country",width=20,font=("bold", 10))

label\_4.place(x=70,y=340)

list1 = ['Canada','India','UK','Nepal','Iceland','South Africa'];

droplist=OptionMenu(root,c, \*list1)

droplist.config(width=15)

c.set('select your country')

droplist.place(x=240,y=340)

Button(root, text='Submit',width=20,bg='brown',fg='white',command=database).place(x=180,y=405)

Button(root, text='Login',width=20,bg='brown',fg='white',command=login).place(x=180,y=435)

root.mainloop()

**USER LOGIN:**

from tkinter import \*

import sqlite3

import os

from tkinter import messagebox

root = Tk()

root.geometry('600x400')

root.title("LOGIN FORM")

UserName = StringVar()

PassWord = StringVar()

def cl2():

os.system("SignUpUser.py")

def cl():

nm = UserName.get()

ps = PassWord.get()

with sqlite3.connect("Form.db") as db:

cursor = db.cursor()

find\_user = ("SELECT \* FROM Student WHERE FullName = ? AND PassWord = ?")

cursor.execute(find\_user,[(nm),(ps)])

result = cursor.fetchall()

if(nm=="" and ps==""):

messagebox.showerror("Error","INVALID USERNAME Or PASSWORD")

if result:

os.system("BookingRequest.py")

else:

messagebox.showerror("Error","INVALID USERNAME Or PASSWORD")

label\_0 = Label(root, text="USER LOGIN",width=20,font=("bold", 20))

label\_0.place(y=30)

label\_1 = Label(root, text="USERNAME",width=20,font=("bold", 15))

label\_1.place(x=80,y=130)

entry\_1 = Entry(root,textvar=UserName)

entry\_1.place(x=270,y=135)

label\_2 = Label(root, text="PASSWORD",width=20,font=("bold", 15))

label\_2.place(x=80,y=180)

entry\_2 = Entry(root,textvar=PassWord)

entry\_2.place(x=270,y=185)

Button(root, text='Login',width=25,bg='brown',fg='white',command=cl).place(x=240,y=250)

Button(root, text='SignUp',width=25,bg='brown',fg='white',command=cl2).place(x=240,y=290)

root.mainloop()

**DRIVER SIGNUP:**

from tkinter import \*

import sqlite3

import random

import string

import os

root = Tk()

root.geometry('500x500')

root.title("Driver Registration Form")

def login():

os.system("DriverLogin.py")

Fullname=StringVar()

Email=StringVar()

Password = StringVar()

MobileNo=StringVar()

var = IntVar()

c=StringVar()

def database():

name1=Fullname.get()

mob1 = MobileNo.get()

email=Email.get()

gender=var.get()

country=c.get()

password = Password.get()

nm1 = str(random.randint(1,101))

st1 = random.choice(string.ascii\_letters)

st2 = random.choice(string.ascii\_letters)

st3 = random.choice(string.ascii\_letters)

st4 = random.choice(string.ascii\_letters)

nm2 = str(random.randint(1,101))

nm3 = str(random.randint(1,101))

nm4 = str(random.randint(1,101))

st5 = random.choice(string.ascii\_letters)

st6 = random.choice(string.ascii\_letters)

st7 = random.choice(string.ascii\_letters)

st8 = random.choice(string.ascii\_letters)

res = nm1+st1+st2+st3+st4+nm2+nm3+nm4+st5+st6+st7+st8

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

with conn:

cursor=conn.cursor()

cursor.execute('CREATE TABLE IF NOT EXISTS Driver (ID INTEGER PRIMARY KEY AUTOINCREMENT,Dr\_ID int,Fullname TEXT UNIQUE,PassWord TEXT UNIQUE,Mobile TEXT,Email TEXT,Gender TEXT,country TEXT)')

cursor.execute('INSERT INTO Driver (Dr\_ID,FullName,PassWord,Mobile,Email,Gender,country) VALUES(?,?,?,?,?,?,?)',(res,name1,password,mob1,email,gender,country))

conn.commit()

label\_0 = Label(root, text="NEW DRIVER",width=20,font=("bold", 20))

label\_0.place(x=90,y=53)

label\_1 = Label(root, text="FullName",width=20,font=("bold", 10))

label\_1.place(x=80,y=130)

entry\_1 = Entry(root,textvar=Fullname)

entry\_1.place(x=240,y=130)

label\_pass = Label(root, text="PassWord",width=20,font=("bold", 10))

label\_pass.place(x=80,y=165)

entry\_pass = Entry(root,textvar=Password)

entry\_pass.place(x=240,y=165)

label\_no = Label(root,text = "Mobile no.",width=20,font=("bold",10))

label\_no.place(x=80,y=203)

entry\_no = Entry(root,textvar=MobileNo)

entry\_no.place(x=240,y=203)

label\_2 = Label(root, text="Email",width=20,font=("bold", 10))

label\_2.place(x=68,y=240)

entry\_2 = Entry(root,textvar=Email)

entry\_2.place(x=240,y=240)

label\_3 = Label(root, text="Gender",width=20,font=("bold", 10))

label\_3.place(x=70,y=290)

Radiobutton(root, text="Male",padx = 5, variable=var, value=1).place(x=235,y=290)

Radiobutton(root, text="Female",padx = 20, variable=var, value=2).place(x=290,y=290)

label\_4 = Label(root, text="country",width=20,font=("bold", 10))

label\_4.place(x=70,y=340)

list1 = ['Canada','India','UK','Nepal','Iceland','South Africa'];

droplist=OptionMenu(root,c, \*list1)

droplist.config(width=15)

c.set('select your country')

droplist.place(x=240,y=340)

Button(root, text='Submit',width=20,bg='brown',fg='white',command=database).place(x=180,y=405)

Button(root, text='Login',width=20,bg='brown',fg='white',command=login).place(x=180,y=445)

root.mainloop()

**DRIVER LOGIN:**

from tkinter import \*

import sqlite3

import os

from tkinter import messagebox

root = Tk()

root.geometry('600x400')

root.title("DRIVER LOGIN FORM")

UserName = StringVar()

PassWord = StringVar()

def cl2():

os.system("DriverSignUp.py")

def cl():

nm = UserName.get()

ps = PassWord.get()

with sqlite3.connect("DriverForm.db") as db:

cursor = db.cursor()

find\_user = ("SELECT \* FROM Driver WHERE FullName = ? AND PassWord = ?")

cursor.execute(find\_user,[(nm),(ps)])

result = cursor.fetchall()

if(nm=="" and ps==""):

messagebox.showerror("Error","INVALID USERNAME Or PASSWORD")

elif result:

for i in result:

os.system("DriverInterface.py")

else:

messagebox.showerror("Error","INVALID USERNAME Or PASSWORD")

label\_0 = Label(root, text="DRIVER LOGIN",width=20,font=("bold", 20))

label\_0.place(y=30)

label\_1 = Label(root, text="USERNAME",width=20,font=("bold", 15))

label\_1.place(x=80,y=130)

entry\_1 = Entry(root,textvar=UserName)

entry\_1.place(x=270,y=135)

label\_2 = Label(root, text="PASSWORD",width=20,font=("bold", 15))

label\_2.place(x=80,y=180)

entry\_2 = Entry(root,textvar=PassWord)

entry\_2.place(x=270,y=185)

Button(root, text='Login',width=25,bg='brown',fg='white',command=cl).place(x=240,y=250)

Button(root, text='SignUp',width=25,bg='brown',fg='white',command=cl2).place(x=240,y=280)

root.mainloop()

**BOOKING REQUEST:**

from tkinter import \*

import sqlite3

root = Tk()

root.geometry('500x500')

root.title("BOOKING REQUEST")

MobileNo = StringVar()

from\_ = StringVar()

to\_ = StringVar()

Day = StringVar()

Time = StringVar()

i = 0

j = 0

k = 0

fr = [[0,5,5,5,5,10,10,10,20,15,12],

[5,0,2,2,5,10,10,10,25,22,17],

[5,2,0,2,2,10,10,10,30,25,2],

[15,7,2,0,2,20,20,20,35,30,10],

[5,10,15,15,0,15,15,15,30,27,32],

[10,15,25,25,20,0,5,5,15,20,30],

[10,15,25,25,20,5,0,5,20,15,23],

[10,15,25,25,20,10,15,0,17,32,22],

[20,25,30,30,27,15,15,15,0,10,35],

[11,15,37,40,11,15,30,15,22,0,33],

[12,22,33,12,20,25,17,18,19,14,0]]

def database(fr2):

global i,j;

mobileNo = MobileNo.get()

fr = from\_.get()

to = to\_.get()

day = Day.get()

time = Time.get()

location = from\_.get()

destination = to\_.get()

if(location=='BH1'):

i = 0

elif(location=='BH4'):

i = 1

elif(location=='BH5'):

i = 2

elif(location=='BH6'):

i = 3

elif(location=='BH7'):

i = 4

elif(location=='GH1'):

i = 5

elif(location=='GH2'):

i = 6

elif(location=='UNI\_POLIS'):

i = 7

elif(location=='LPU-GATE'):

i = 8

elif(location=='BLOCK-27'):

i = 9

elif(location=='BLOCK-55'):

i = 10

if(destination=='BH1'):

j = 0

elif(destination=='BH4'):

j = 1

elif(destination=='BH5'):

j = 2

elif(destination=='BH6'):

j = 3

elif(destination=='BH7'):

j = 4

elif(destination=='GH1'):

j = 5

elif(destination=='GH2'):

j = 6

elif(destination=='UNI\_POLIS'):

j = 7

elif(destination=='LPU-GATE'):

j = 8

elif(destination=='BLOCK-27'):

j = 9

elif(destination=='BLOCK-55'):

j = 10

s2 = "Rs. "

s3 = str(fr2[i][j])

result = s2+s3

label\_charges = Label(root, text="Fare",width=20,font=("bold", 10))

label\_charges.place(x=70,y=280)

label\_fare = Label(root,text=result)

label\_fare.place(x=240,y=280)

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

with conn:

cursor=conn.cursor()

cursor.execute('CREATE TABLE IF NOT EXISTS OrderBooking(ID INTEGER PRIMARY KEY AUTOINCREMENT,MobileNo TEXT,Location TEXT,Destination TEXT ,Day TEXT,Time TEXT)')

cursor.execute('INSERT INTO OrderBooking(MobileNo,Location,Destination,Day,Time) VALUES(?,?,?,?,?)',(mobileNo,fr,to,day,time))

conn.commit()

label\_0 = Label(root, text="Booking Request",width=20,font=("bold", 20))

label\_0.place(x=90,y=53)

label\_1 = Label(root, text="MobileNo",width=20,font=("bold", 10))

label\_1.place(x=80,y=130)

entry\_1 = Entry(root,textvar=MobileNo)

entry\_1.place(x=240,y=130)

label\_4 = Label(root, text="FROM",width=20,font=("bold", 10))

label\_4.place(x=70,y=180)

list1 = ['BH1','BH4','BH5','BH6','BH7','GH1','GH2','UNI\_POLIS','LPU-GATE','BLOCK-27','BLOCK-55'];

droplist=OptionMenu(root,from\_, \*list1)

droplist.config(width=30)

from\_.set('SELECT YOUR LOCATION')

droplist.place(x=240,y=180)

label\_5 = Label(root, text="TO",width=20,font=("bold", 10))

label\_5.place(x=70,y=230)

list1 = ['BH1','BH2','BH3','BH4','BH5','BH6','UNI\_POLIS','LPU-GATE','LAW-GATE','BLOCK-27'];

droplist=OptionMenu(root,to\_, \*list1)

droplist.config(width=30)

to\_.set('SELECT YOUR DESTINATION')

droplist.place(x=240,y=230)

label\_day = Label(root, text="Day",width=20,font=("bold", 10))

label\_day.place(x=70,y=320)

entry\_day = Entry(root,textvar=Day)

entry\_day.place(x=240,y=320)

label\_time = Label(root, text="Time",width=20,font=("bold", 10))

label\_time.place(x=70,y=370)

entry\_time = Entry(root,textvar=Time)

entry\_time.place(x=240,y=370)

Button(root, text='Submit',width=20,bg='brown',fg='white',command = lambda:database(fr)).place(x=180,y=405)

root.mainloop()

**DRIVER INTERFACE:**

from tkinter import \*

import sqlite3

root = Tk()

root.geometry('500x500')

root.title("DRIVER INTERFACE")

ID = IntVar()

MobileNo = StringVar()

from\_ = StringVar()

to\_ = StringVar()

Day = StringVar()

Time = StringVar()

a = 0

b = 0

fr = [[0,5,5,5,5,10,10,10,20,15,12],

[5,0,2,2,5,10,10,10,25,22,17],

[5,2,0,2,2,10,10,10,30,25,2],

[15,7,2,0,2,20,20,20,35,30,10],

[5,10,15,15,0,15,15,15,30,27,32],

[10,15,25,25,20,0,5,5,15,20,30],

[10,15,25,25,20,5,0,5,20,15,23],

[10,15,25,25,20,10,15,0,17,32,22],

[20,25,30,30,27,15,15,15,0,10,35],

[11,15,37,40,11,15,30,15,22,0,33],

[12,22,33,12,20,25,17,18,19,14,0]]

with sqlite3.connect("Booking.db ") as db:

cursor = db.cursor()

cursor.execute("SELECT \* FROM OrderBooking WHERE ID = (SELECT MAX(ID) FROM OrderBooking)")

result = cursor.fetchall()

if(result):

for i in result:

ID = i[0]

MobileNo = i[1]

from\_ = i[2]

to\_ = i[3]

Day = i[4]

Time = i[5]

else:

print("No Booking Found")

if(from\_=='BH1'):

a = 0

elif(from\_=='BH4'):

a = 1

elif(from\_=='BH5'):

a = 2

elif(from\_=='BH6'):

a = 3

elif(from\_=='BH7'):

a = 4

elif(from\_=='GH1'):

a = 5

elif(from\_=='GH2'):

a = 6

elif(from\_=='UNI\_POLIS'):

a = 7

elif(from\_=='LPU-GATE'):

a = 8

elif(from\_=='BLOCK-27'):

a = 9

elif(from\_=='BLOCK-55'):

a = 10

if(to\_=='BH1'):

b = 0

elif(to\_=='BH4'):

b = 1

elif(to\_=='BH5'):

b = 2

elif(to\_=='BH6'):

b = 3

elif(to\_=='BH7'):

b = 4

elif(to\_=='GH1'):

b = 5

elif(to\_=='GH2'):

b = 6

elif(to\_=='UNI\_POLIS'):

b = 7

elif(to\_=='LPU-GATE'):

b = 8

elif(to\_=='BLOCK-27'):

b = 9

elif(to\_=='BLOCK-55'):

b = 10

s2 = "Rs. "

s3 = str(fr[a][b])

result = s2+s3

label\_charges = Label(root, text="Fare",width=20,font=("bold", 10))

label\_charges.place(x=70,y=280)

label\_fare = Label(root,text=result)

label\_fare.place(x=240,y=280)

label\_0 = Label(root, text="Driver Interface",width=20,font=("bold", 20))

label\_0.place(x=90,y=53)

label\_1 = Label(root, text="MobileNo",width=20,font=("bold", 10))

label\_1.place(x=80,y=130)

label\_no = Label(root,text=MobileNo)

label\_no.place(x=240,y=130)

label\_4 = Label(root, text="LOCATION",width=20,font=("bold", 10))

label\_4.place(x=70,y=180)

label\_loc = Label(root,text=from\_)

label\_loc.place(x=240,y=180)

label\_5 = Label(root, text="DESTINATION",width=20,font=("bold", 10))

label\_5.place(x=70,y=230)

label\_dest = Label(root,text=to\_)

label\_dest.place(x=240,y=230)

label\_day = Label(root, text="Day",width=20,font=("bold", 10))

label\_day.place(x=70,y=320)

entry\_day = Label(root,text = Day)

entry\_day.place(x=240,y=320)

label\_time = Label(root, text="Time",width=20,font=("bold", 10))

label\_time.place(x=70,y=370)

entry\_time = Label(root,text=Time)

entry\_time.place(x=240,y=370)

def on():

with sqlite3.connect("Booking.db ") as db:

cursor = db.cursor()

cursor.execute("DELETE FROM OrderBooking WHERE ID = (SELECT MAX(ID) FROM OrderBooking)")

db.commit()

cursor.close()

label\_no.config(text="-----")

label\_loc.config(text="-----")

label\_dest.config(text="-----")

entry\_day.config(text="-----")

entry\_time.config(text="-----")

label\_fare.config(text="-----")

Button(root, text='ON THE WAY',width=20,command = on,bg='brown',fg='white').place(x=180,y=405)

root.mainloop()

**RESULTS:**

In the Previous System,Details are Stored Manually in papers,to share the details between employees was a Financial drawback. Updations in the details is a tedious task.

But this system was proposed to overcome the above drawbacks.

Functionalities and advantages of proposed system are:

* Data is Centralized which has overcome the Sharing problem in previous system.
* As data is Maintained electronically, it’s easy for cab booking management system a person to update the details, which has overcome the tedious updation in previous system.
* Maintenance is easy and performance is good.
* Mainly the system has automated the Transportation Process.

**REFERENCES:**

[**www.techpedia.srishti.org**](http://www.techpedia.srishti.org)

[**www.wikipedia.com**](http://www.wikipedia.com)