

CAR RENTAL BOOKING SYSTEM

MINOR PROJECT REPORT

By

ARAVIND A (RA2211026010554)

RISHIRAM B (RA2211026010553)

SURYA SATHYANARAYANAN(RA2211026010562)

Under the guidance of

Dr.Anousouya Devi

In partial fulfilment for the Course

of

21CSC203P – ADVANCED PROGRAMMING PRACTICE

in CINTEL-AI/ML



FACULTY OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTING

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR

NOVEMBER 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

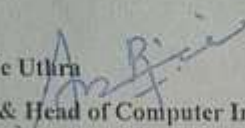
BONAFIDE CERTIFICATE

Certified that this minor project report for the course **21CSC203P ADVANCED PROGRAMMING PRACTICE** entitled in "Car Rental Booking System" is the bonafide work of Aravind A (RA2211026010554) , Rishiram B(RA2211026010553) and Surya Sathyanarayanan(RA2211026010562) who carried out the work under my supervision.



Dr. Anousouya Devi
Assistant Professor
CINTEL
SRM Institute of Science and Technology
Kattankulathur

SIGNATURE



Dr. R Annie Uthra
Professor & Head of Computer Intelligence
School of computing
SRM Institute of Science and Technology
Kattankulathur

ABSTRACT

This Python-based Car Rental Booking System project introduces an efficient and user-friendly platform for streamlining the car rental process. The system aims to automate booking procedures, enhance inventory management, and provide customers with an intuitive interface for seamless interactions. Motivated by the desire to modernize and simplify car rentals, the project addresses challenges associated with manual processes, limited automation, and inventory tracking issues. By leveraging Python's capabilities, the system contributes to a more accessible, error-resistant, and customer-centric car rental experience. Overall, the Car Rental Booking System project seeks to optimize rental operations, providing a reliable and efficient solution for both service providers and customers.

ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors. We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy**, for his encouragement.

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V. Gopal**, for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman**, for imparting confidence to complete my course project

We wish to express my sincere thanks to **Course Audit Professors Dr. Vadivu. G, Professor, Department of Data Science and Business Systems and Dr. Sasikala. E Professor, Department of Data Science and Business Systems and Course Coordinators** for their constant encouragement and support.

We are highly thankful to our my Course project Faculty **Dr. Anousouya Devi, Assistant Professor, CINTEL** for her assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our **Annie Uthra, Head of Computer Intelligence, CINTEL** and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

TABLE OF CONTENTS

CHAPTER NO	CONTENTS	PAGE NO
1	INTRODUCTION	1-2
2	REQUIREMENTS	3
3	GUI DESIGN AND DB CONNECTIVITY	4
4	IMPLEMENTATION	6
5	RESULT	9
6	CONCLUSION	11
7	REFERENCES	12

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

The motivation behind our Car Rental Booking System stems from a commitment to redefine the travel experience. Recognizing the evolving needs of modern consumers, we aim to eliminate the hassles traditionally associated with renting a vehicle. Our project is driven by a desire to offer unparalleled convenience, empowering users to effortlessly access and book a diverse range of vehicles online. By prioritizing user-friendly interfaces, real-time information, and rigorous safety measures, we aspire to revolutionize the way people plan and execute their journeys. This project is not just about transportation; it's about providing freedom, choice, and peace of mind to individuals seeking a reliable and enjoyable travel experience.

1.2 OBJECTIVE

The primary objective of the Car Rental Booking System project is to develop a user-friendly and efficient platform for managing car rentals. The specific objectives include:

- **User-Friendly Interface:** Create an intuitive interface for customers to browse available cars, check rental details, and make bookings effortlessly.
- **Automated Booking Process:** Implement a system that automates the booking process, from selecting a vehicle to generating invoices, reducing manual efforts and minimizing errors.
- **Inventory Management:** Develop a robust inventory management system to keep track of available cars, ensuring accurate and real-time information for both customers and administrators.

1.3 PROBLEM STATEMENT

Existing car rental systems frequently face challenges related to manual data entry, limited automation, and a lack of real-time inventory tracking. These issues can lead to errors, delays, and an overall suboptimal experience for both customers and service providers. The Car Rental Booking System project seeks to address these problems by introducing a comprehensive, automated solution that improves the efficiency of the booking process, minimizes errors, and enhances overall customer satisfaction. Through the implementation of this Python-based system, we aim to provide a reliable and modern alternative to conventional car rental management.

1.4 CHALLENGES

The Car Rental Booking System project faces challenges in implementing a seamless and secure online platform, ensuring real-time data synchronization for accurate availability, and establishing robust cybersecurity measures to protect customer information. Integrating a diverse fleet while maintaining consistent vehicle maintenance poses logistical hurdles. Additionally, addressing customer support needs and building user trust amid concerns about transparency and hidden costs demands careful attention. Adapting to evolving technology, scalability requirements, and potential regulatory constraints further present challenges. Successfully navigating these complexities is crucial to delivering a reliable, transparent, and user-friendly car rental solution.

CHAPTER 2

REQUIREMENTS

2.1 User Interface:

- Develop an intuitive interface for customers to browse cars and complete bookings seamlessly.
- Ensure responsiveness for a user-friendly experience across various devices.

2.1 Automated Booking Process:

- Implement an automated system for vehicle selection, booking confirmation, and invoice generation.
- Secure and user-friendly payment processing for reservations.

2.3 Inventory Management:

- Create a centralized database for real-time updates on available cars.
- Provide admin tools for adding, removing, or updating vehicles.

2.4 Security Measures:

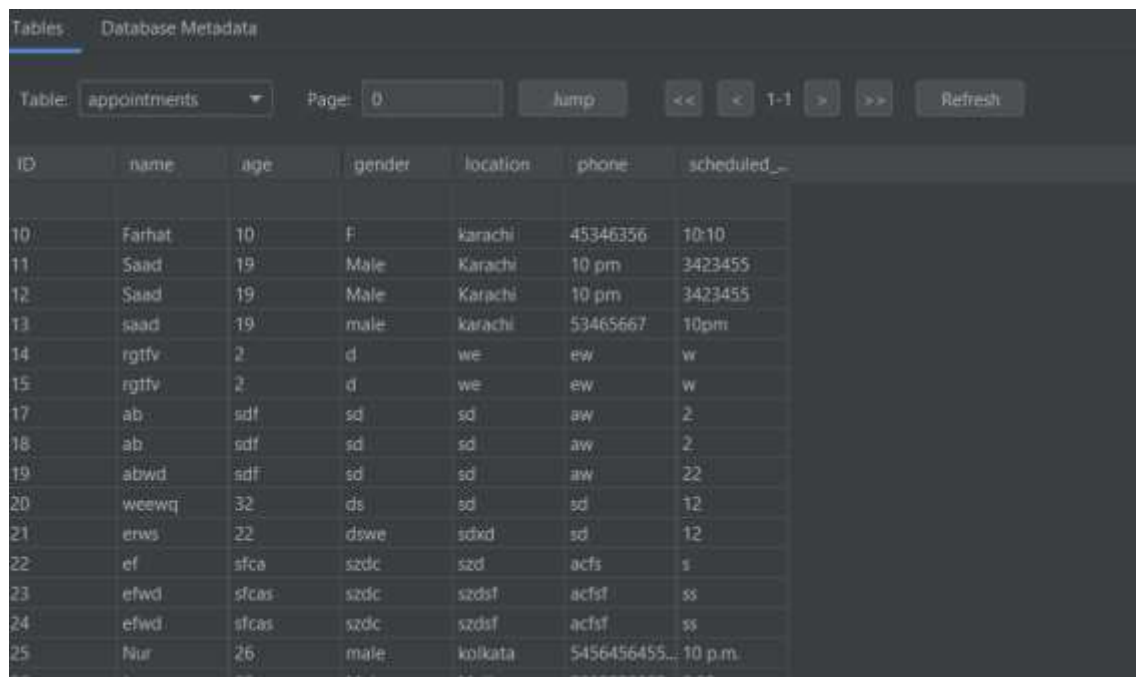
- Implement encryption for user and payment data security.
- Regularly update and patch vulnerabilities to protect against security threats.

CHAPTER 3

GUI DESIGN AND DB CONNECTIVITY

3.1 DATABASE CONNECTIVITY

The Booking of the Users should be Stored on the Database. We used the **MySQL** to store the Booking. The Mysql Database then can be Used to check the Booking Details and Allocate the Car. Database Connectivity is the key difference maker between Normal Voting system and Online Voting System. Here is the EXAMPLE of a database:

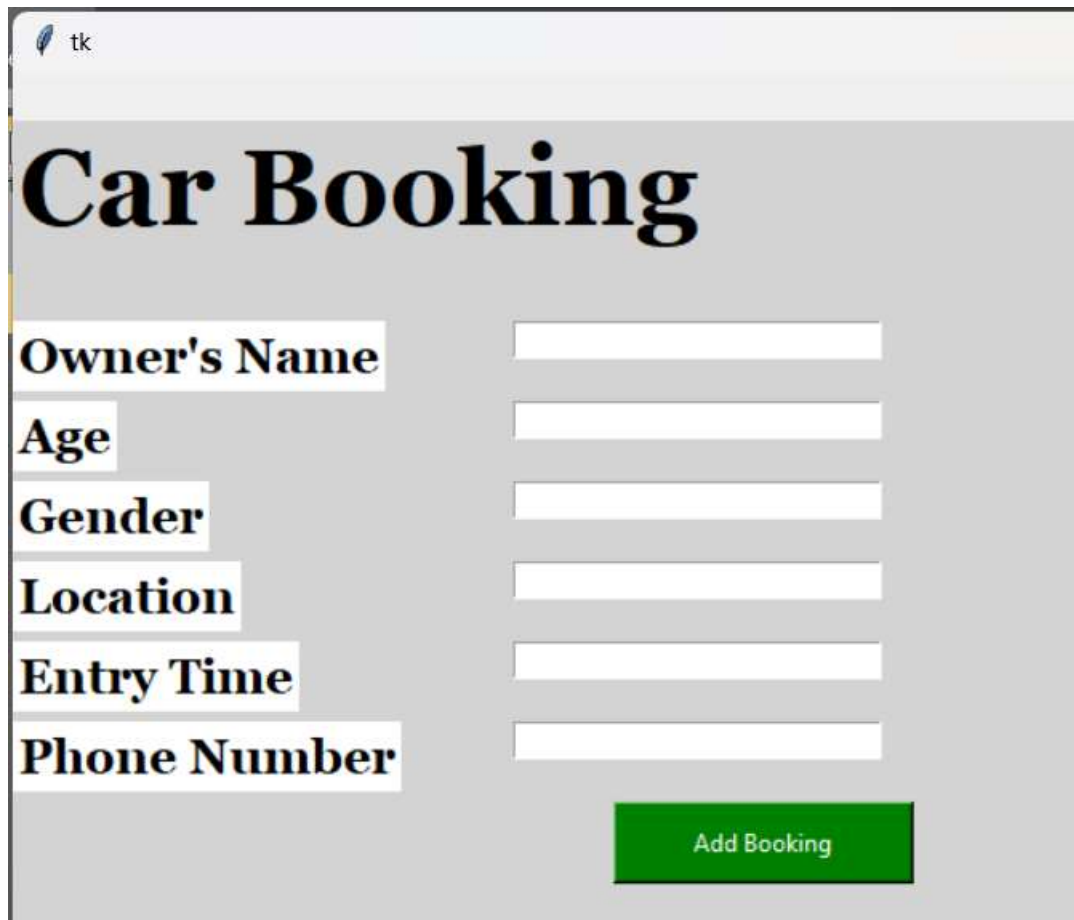


ID	name	age	gender	location	phone	scheduled_
10	Faihat	10	F	karachi	45346356	10:10
11	Saad	19	Male	Karachi	10 pm	3423455
12	Saad	19	Male	Karachi	10 pm	3423455
13	saad	19	male	karachi	53465667	10pm
14	rgttv	2	d	we	ew	w
15	rgttv	2	d	we	ew	w
17	ab	sdf	sd	sd	aw	2
18	ab	sdf	sd	sd	aw	2
19	abwd	sdf	sd	sd	aw	22
20	weewq	32	ds	sd	sd	12
21	erws	22	dswe	sdxd	sd	12
22	ef	sfca	szdc	szd	acfs	s
23	efwd	sfcas	szdc	szdsf	acfsf	ss
24	efwd	sfcas	szdc	szdsf	acfsf	ss
25	Nur	26	male	kolkata	5456456455...	10 p.m.

Figure 3.1.1 (Image from MySQL Workbench.)

3.2 GUI [Graphical-User Interface]

As the project is made in Python, by the help of tKinter We have created the User Interface for Users to Give their Votes. GUI as Follows:



The image shows a Tkinter window titled "Car Booking". The window has a title bar with a feather icon and the text "tk". The main content area has a large, bold, black title "Car Booking". Below the title, there are six labels: "Owner's Name", "Age", "Gender", "Location", "Entry Time", and "Phone Number". Each label is followed by a text input field. At the bottom right of the window, there is a green button with the text "Add Booking".

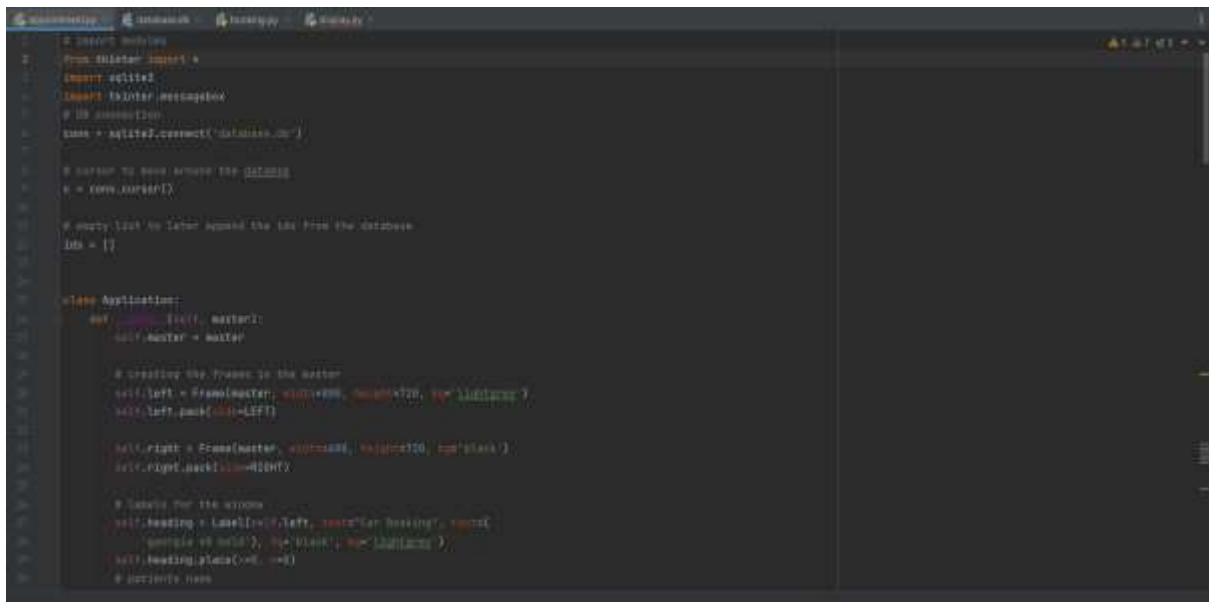
Figure 3.2.1 (Example of GUI From User UI)

CHAPTER 4

IMPLEMENTATION

The Car Rental Booking System implemented in Python on PyCharm streamlines the vehicle reservation process. Utilizing Python's versatility, the backend employs frameworks like Flask or Django for robust server-side logic. PyCharm's integrated development environment enhances code efficiency and readability. The system features a user-friendly command-line or graphical interface, allowing customers to browse available vehicles, make reservations, and view booking details. Integration with databases, such as SQLite or MySQL, ensures data persistence.

Using the **PyCharm**, Java code have been Implemented As follows:

A screenshot of the PyCharm IDE interface. The main editor window displays Python code for an application. The code includes imports for tkinter, sqlite3, and messagebox. It defines a database connection and a list of IDs. A class named 'Application' is defined with methods for window setup, including labels, buttons, and a text area. The code is written in a dark-themed editor with syntax highlighting.

```
1 # Import modules
2 from tkinter import *
3 import sqlite3
4 import tkinter.messagebox
5 # DB connection
6 conn = sqlite3.connect('database.db')
7
8 cursor to send across the database
9 c = conn.cursor()
10
11 # query that we later append the ids from the database
12 ids = []
13
14 class Application:
15     def __init__(self, master):
16         self.master = master
17
18         # creating the frame to the master
19         self.left = Frame(master, width=300, height=100, bg='lightblue')
20         self.left.pack(side=LEFT)
21
22         self.right = Frame(master, width=300, height=100, bg='black')
23         self.right.pack(side=RIGHT)
24
25         # labels for the window
26         self.heading = Label(self.left, text="Car Booking", font=
27             'serif 16 bold', bg='black', fg='lightblue')
28         self.heading.pack()
29
30 # create the window
```

Figure 4.1(Source Code of the admin page)

```

56 self.name = label(self, text, text="Name's Name", font=
57     "serif", font_size=12, fg="black", bg="white")
58 self.name.place(x=0, y=0)
59
60 self.val1 = self.text.get()
61 self.val2 = self.phone.get()
62
63 # checking if the user input is empty
64 if self.val1 == "" or self.val2 == "" or self.val3 == "" or self.val4 == "" or self.val5 == "":
65     tkinter.messagebox.showinfo("Warning", "Please fill in all fields")
66     return
67
68 # send me data to the database
69 sql = "INSERT INTO 'apartment' (name, city, phone, location, zip_code, floor, price, sqft, p, v)"
70 o.execute(sql, (self.val1, self.val2, self.val3, self.val4,
71               self.val5, self.val6, self.val7, self.val8))
72
73 conn.commit()
74
75 tkinter.messagebox.showinfo(
76     "Success", "Adding for " + str(self.val1) + " has been created")
77 self.box.insert(END, "Adding floor for " +
78                   str(self.val1) + " at " + str(self.val8))
79
80
81 # creating the object
82 root = Tk()
83 r = Application(root)
84
85 # resolution of the window
86 root.geometry("500x500")
87
88 # connecting the main explore
89 root.mainloop()

```

Figure 4.2

```

1 # create the application
2 from tkinter import *
3 import tkinter.messagebox
4 import sqlite3
5
6 conn = sqlite3.connect('database.db')
7 c = conn.cursor()
8
9
10 class Application:
11     def __init__(self, master):
12         self.master = master
13         # heading label
14         self.heading = Label(master, text="Addings ", font="serif", font_size=12)
15         self.heading.place(x=250, y=100)
16
17         # search criteria -- name
18         self.name = Label(master, text="Enter Name's Name", font="serif", font_size=12)
19         self.name.place(x=0, y=100)
20
21         # entry for the name
22         self.entry1 = Entry(master, width=100)
23         self.entry1.place(x=250, y=100)
24
25         # search button
26         self.search = Button(master, text="Search", width=12, height=1, bg="black", fg="white", command=self.search_db)
27         self.search.place(x=400, y=120)
28
29         # function to search
30         def search_db(self):

```

Figure 4.3 (Source Code of the User page)

```

import sys

# connect to database
conn = sqlite3.connect('hospital.db')
c = conn.cursor()

# empty lists to append later
number = []
patients = []

sql = "SELECT * FROM appointments"
res = c.execute(sql)
for x in res:
    id = x[0]
    name = x[1]
    number.append(id)
    patients.append(name)

# main
class Application:
    def __init__(self, master):
        self.master = master

        self.w = 0

        # heading
        self.heading = Label(master, text="Hospital", font=('arial', 12, 'bold'), bg='green',
                               width=100, height=1)
        self.heading.place(x=0, y=0)

        # button to change heading

```

Figure 4.4

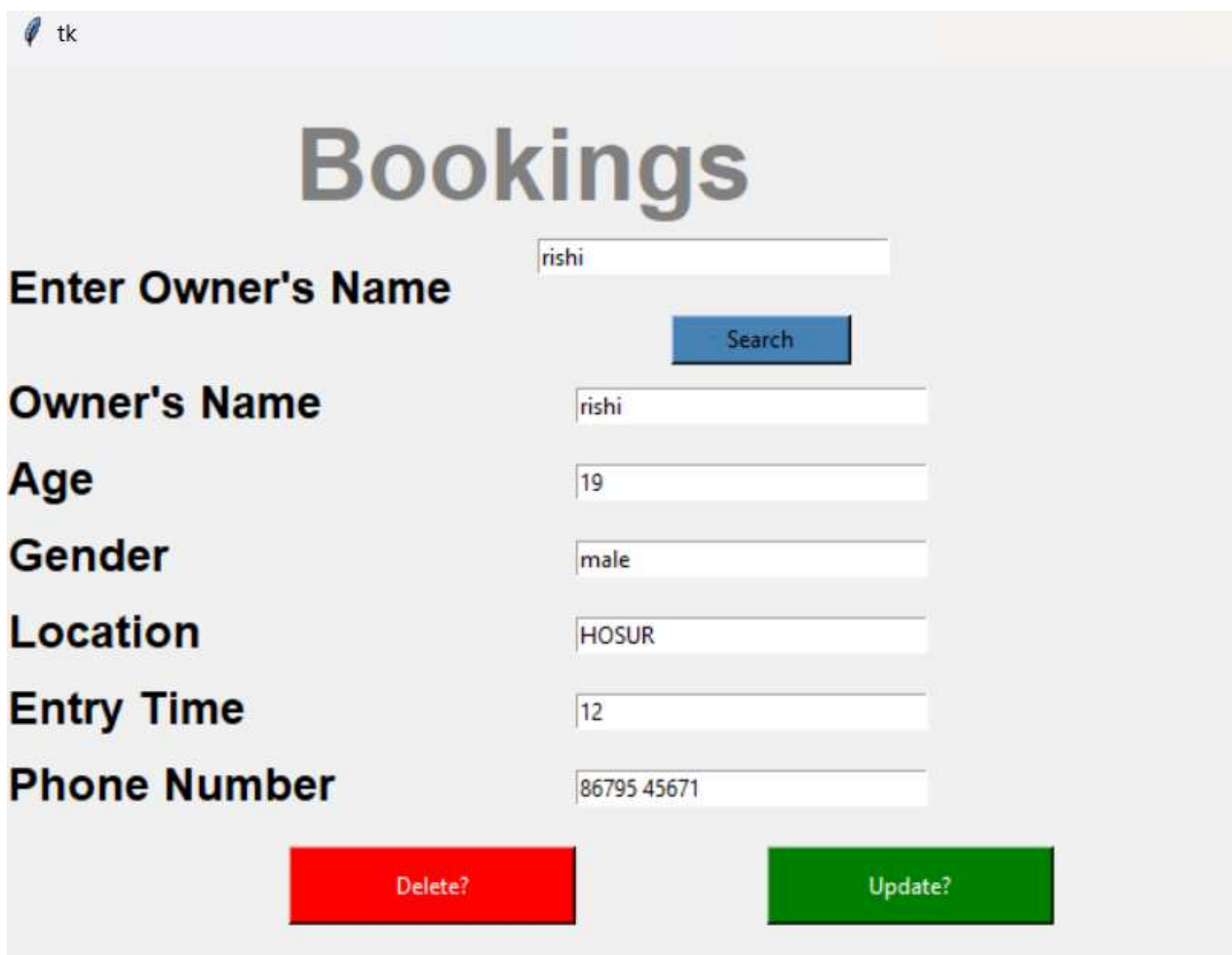
CHAPTER 5

RESULT

5.1 ADMIN UI

The Car Rental Booking System's Admin UI, built with Tkinter in Python on PyCharm, enables administrators to easily search, update, and delete reservations. The Tkinter-based GUI provides an intuitive interface for efficient management, enhancing the user experience and functionality of the project.

Admin Has the Access to Search and Update or Delete the booking Details as Follows:



tk

Bookings

Enter Owner's Name

Owner's Name

Age

Gender

Location

Entry Time

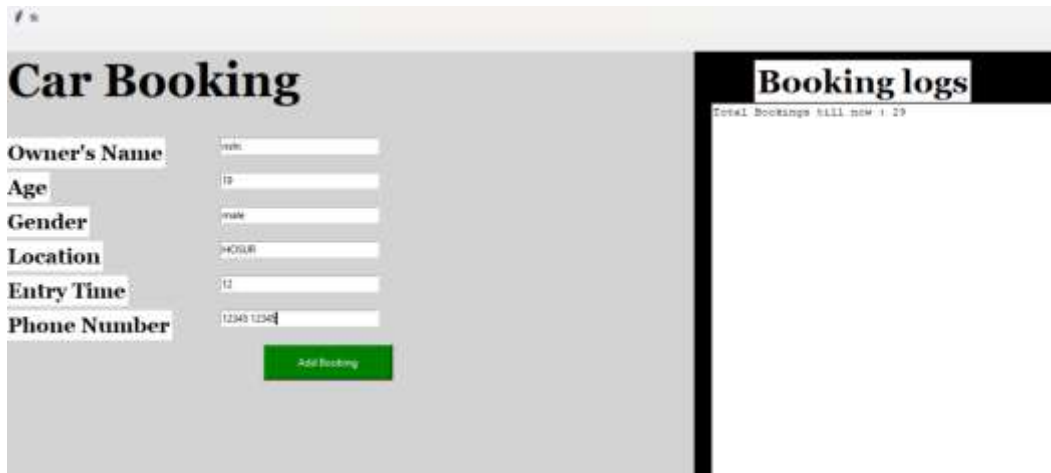
Phone Number

Figure 5.1.1 (Admin UI with update and deletion Option)

5.2 USER UI

The User UI in the Car Rental Booking System, crafted with Tkinter in Python on PyCharm, offers a user-friendly interface for adding information. Users can effortlessly input details, browse available vehicles, and make reservations, enhancing the overall accessibility and simplicity of the booking process.

Users Can Enter their Details for Car booking At the specific time and place. The result as follows:



The screenshot displays a Tkinter window titled "Car Booking". On the left, there is a form with labels and input fields for "Owner's Name", "Age", "Gender", "Location", "Entry Time", and "Phone Number". A green "Add Booking" button is positioned below the form. On the right, a black sidebar contains a white box labeled "Booking logs" with the text "Total Bookings Still now : 29" below it.

Figure 5.2.1 (User UI for Entering the Details)



Figure 5.2.2

CHAPTER 6

CONCLUSION

the Car Rental Booking System project leverages Python's capabilities to address fundamental challenges in the conventional car rental process. By creating an intuitive user interface, automating booking procedures, optimizing inventory management, and prioritizing security measures, the system aims to revolutionize the efficiency and accessibility of car rentals. The project not only streamlines customer interactions but also provides administrators with valuable tools for monitoring and managing operations. With a focus on user satisfaction, automation, and data security, the Car Rental Booking System contributes to the modernization of the car rental industry. As technology continues to evolve, this project stands as a testament to the potential for innovation in simplifying and enhancing everyday services for both consumers and service providers alike.

CHAPTER 7

REFERENCES

- 7.1 <https://docs.python.org/3/library/tkinter.html>
- 7.2 <https://dev.mysql.com/downloads/mysql/>
- 7.3 <https://realpython.com/python-gui-tkinter/>
- 7.4 <https://github.com/topics/car-rental-system?l=python>
- 7.5 <https://www.jetbrains.com/guide/python/tutorials/>
- 7.6 <https://www.javatpoint.com/how-to-connect-database-in-python>
- 7.8 <https://www.w3schools.com/python/>