A PROJECT REPORT ON

“Parking Management System ”

A Mini Project Report Submitted in Partial Fulfilment of Requirement for the 3rd Semester B.TECH Course during the academic year 2022

Submitted by

Name: M.Dinesh Reddy
Registration Number12111718
Name: J.Prane Kumar
Registration Number:12103976
Name: B.Santosh kumar Reddy
Registration Number:12108753

Under the guidance of:

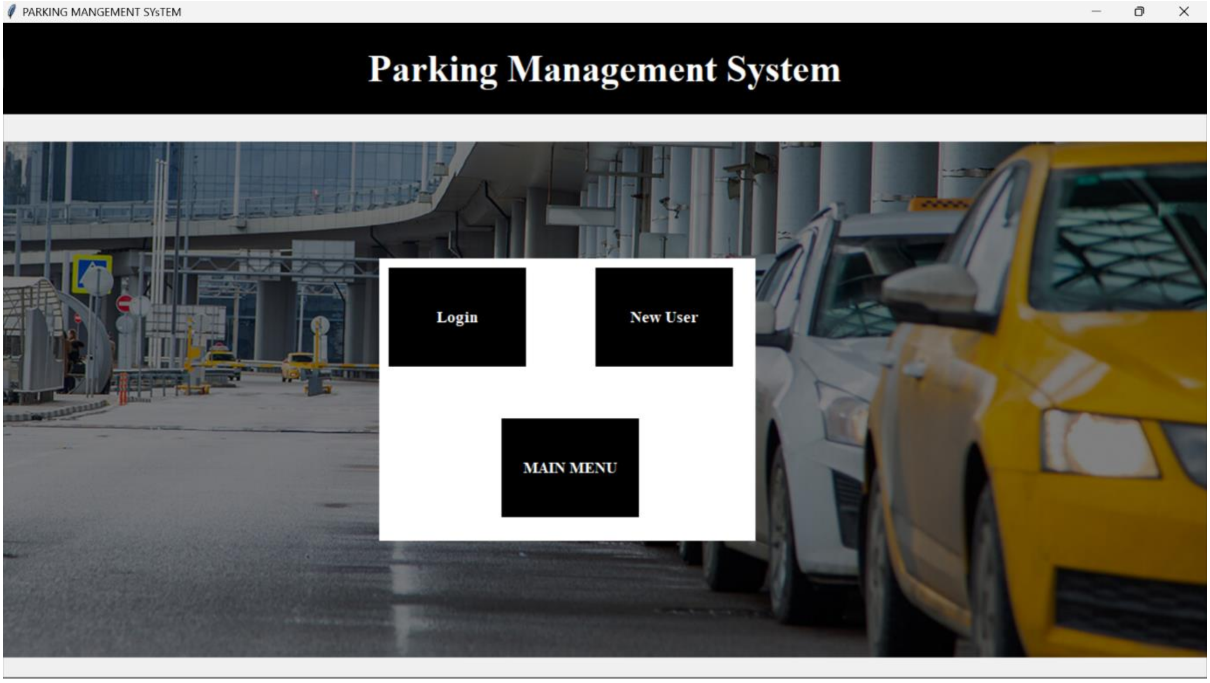
Prof .DR.Deepika Gha

ACKNOWLEDGEMENT

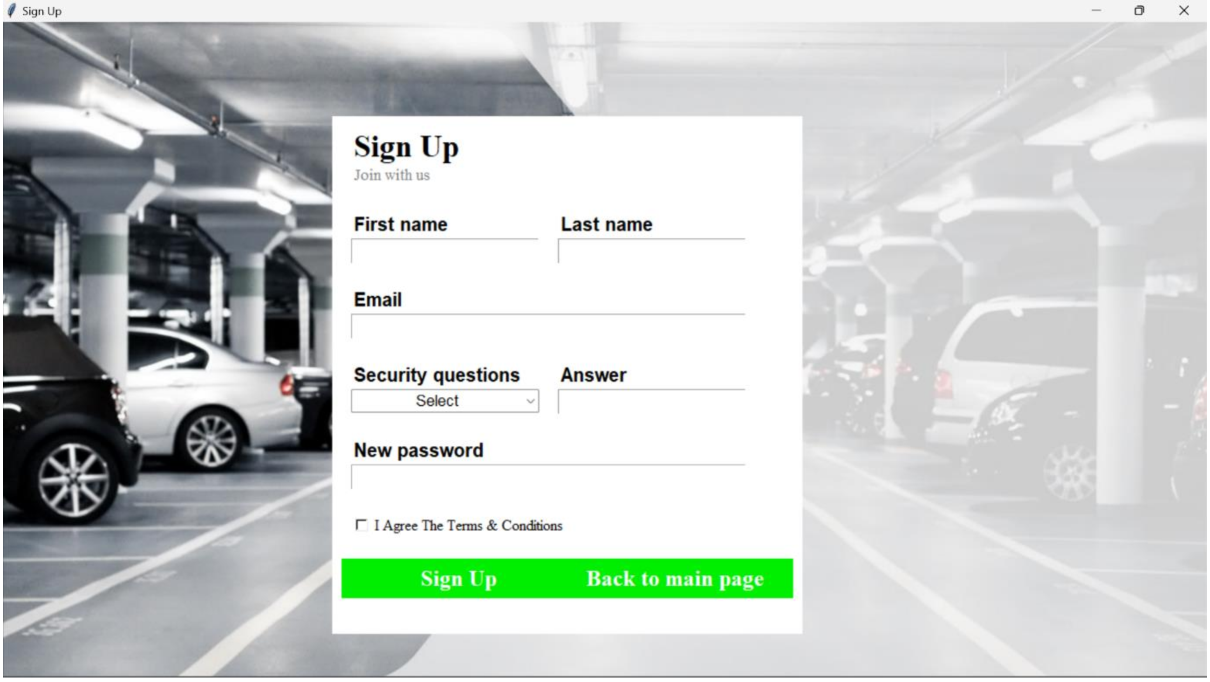
I would like to thank my mentor - Prof. Deepika Ghai for his advice and inputs on this project. Many thanks to my friends and seniors as well, who spent countless hours to listen and provide feedbacks.

SCREENSHOTS

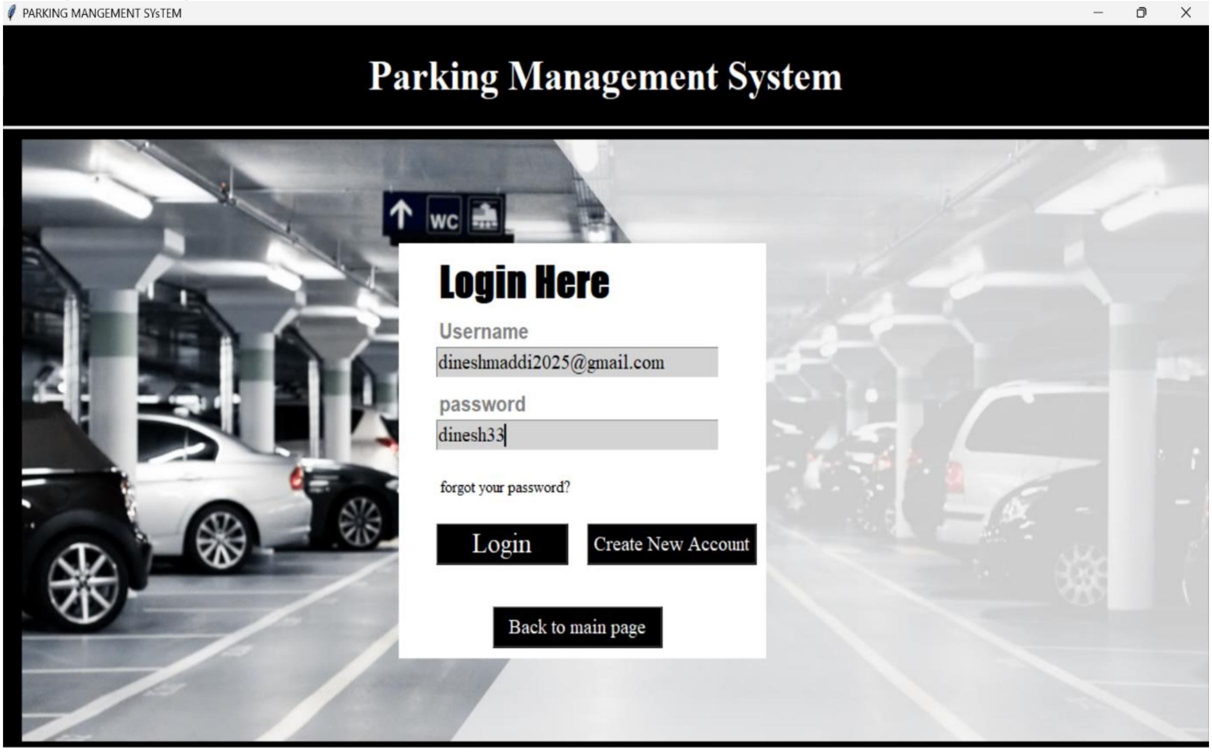
Main page:



SIGN UP PAGE

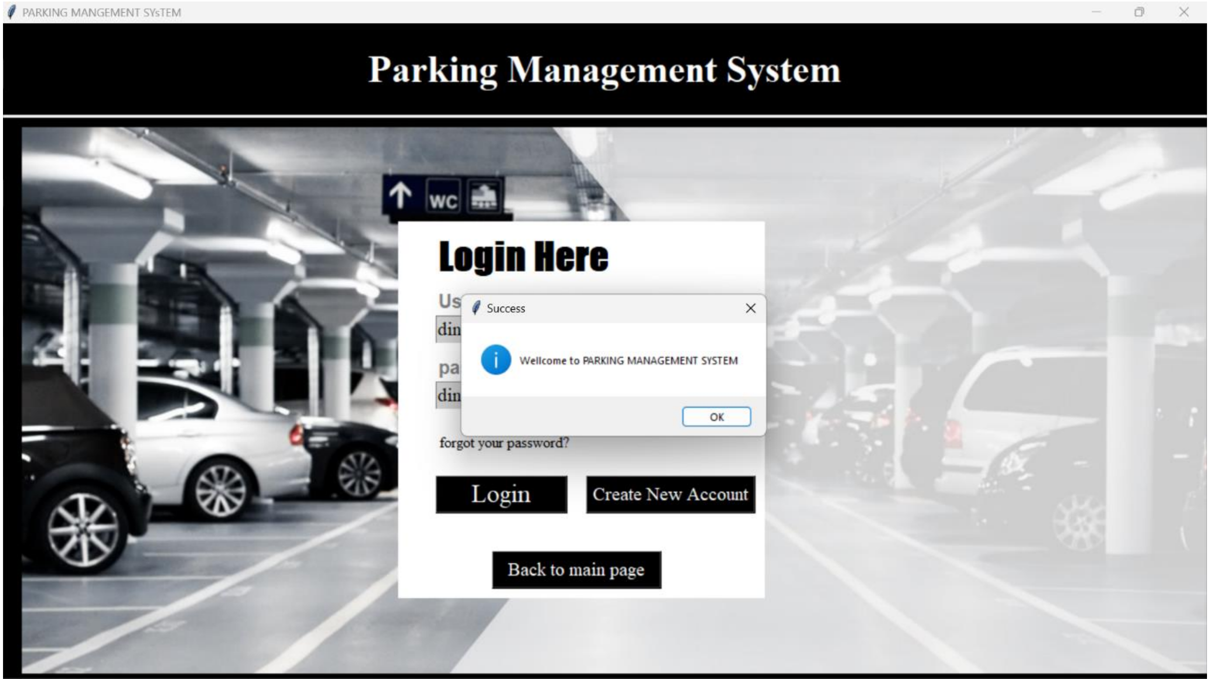


Login page:

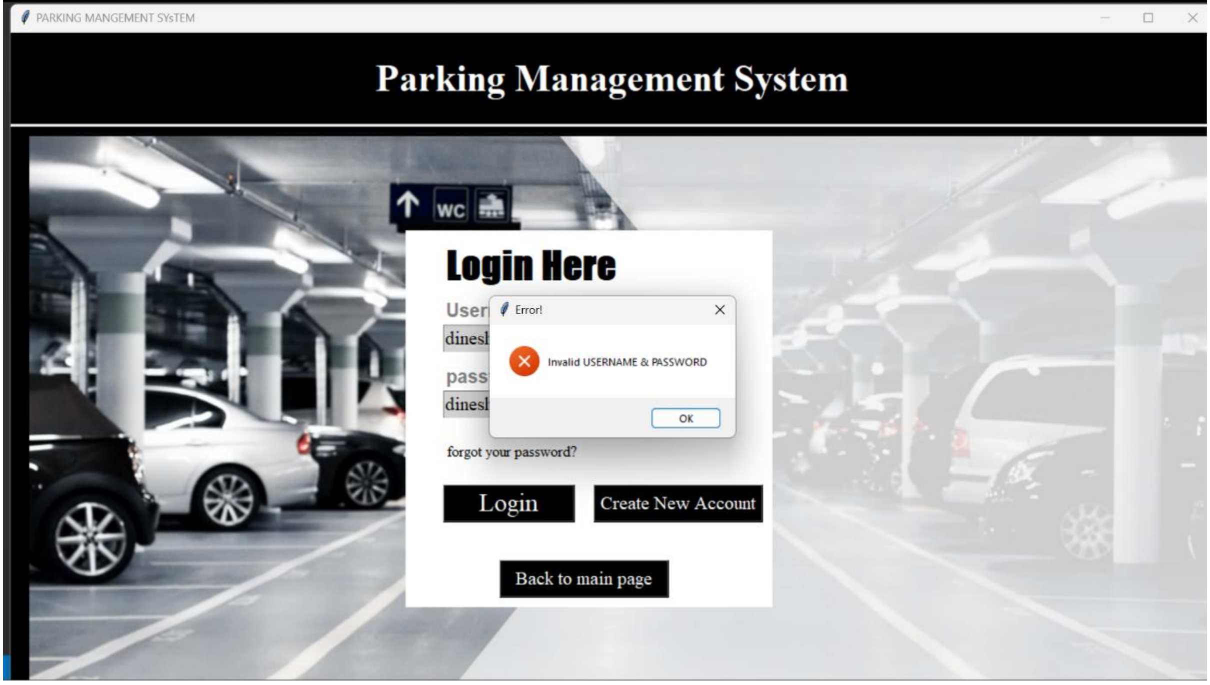


Username
dineshmaddi2025@gmail.com

password
dinesh33

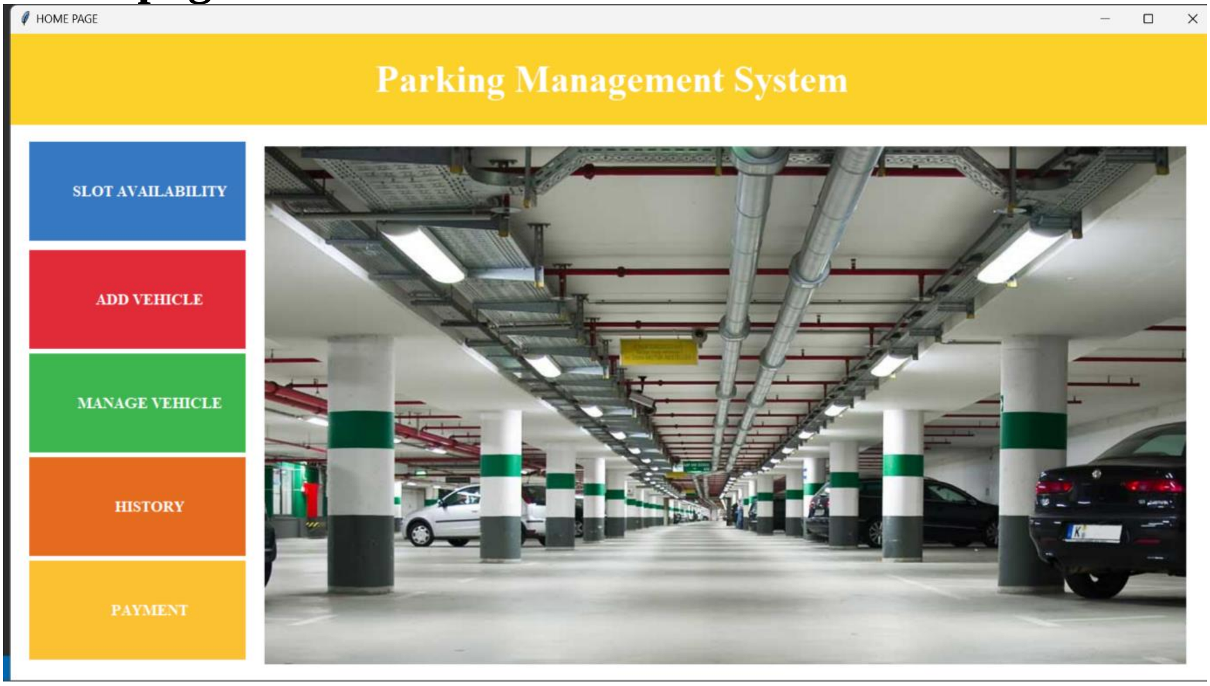


Wellcome to PARKING MANAGEMENT SYSTEM



Invalid USERNAME & PASSWORD

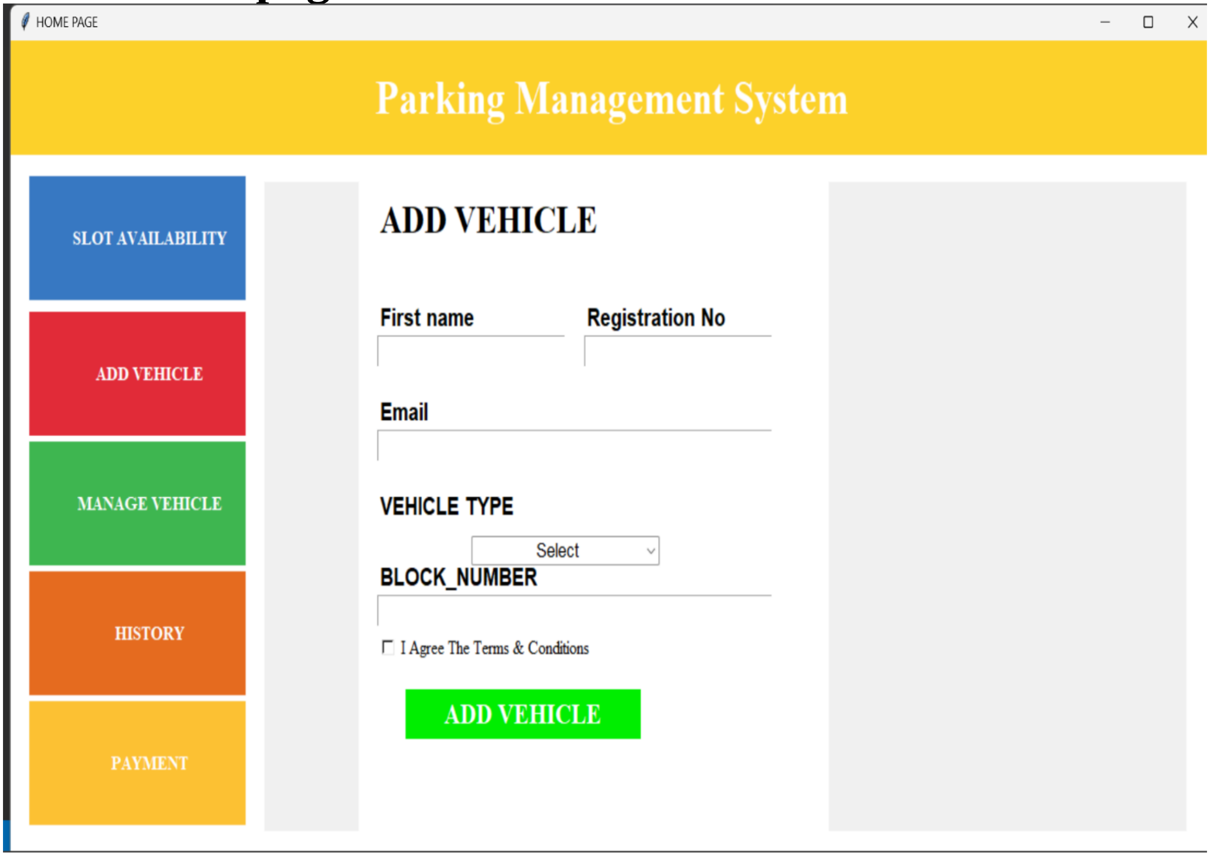
Home page:



HOME PAGE

Parking Management System

Add vehicle page:



HOME PAGE

Parking Management System

ADD VEHICLE

First name

Registration No

Email

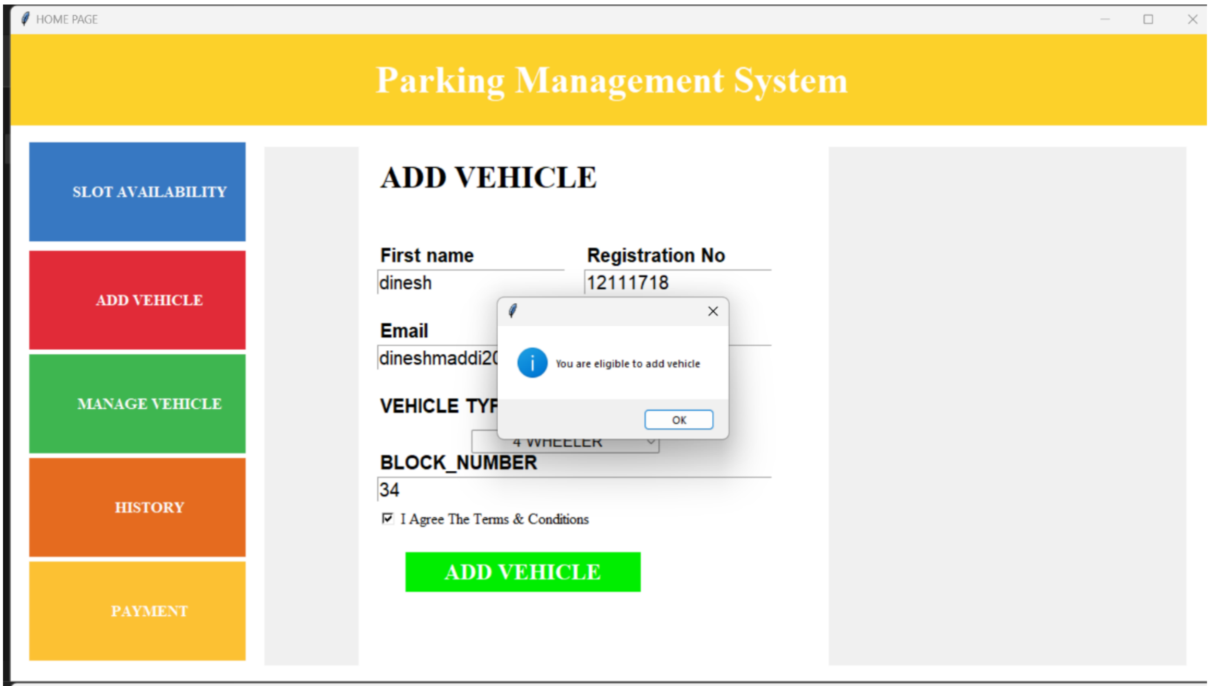
VEHICLE TYPE

Select

BLOCK_NUMBER

□ I Agree The Terms & Conditions

ADD VEHICLE



Parking Management System

ADD VEHICLE

First name: dinesh

Registration No: 12111718

Email: dineshmaddi20

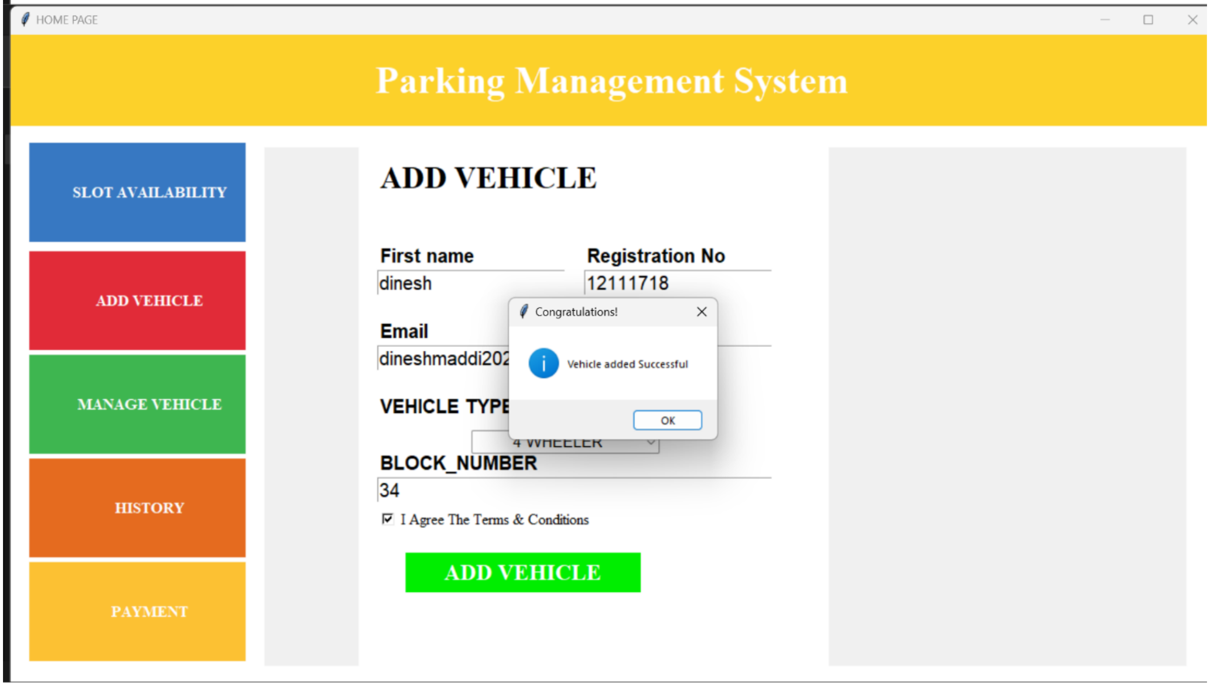
VEHICLE TYP

4 WHEELER

BLOCK_NUMBER: 34

☑ I Agree The Terms & Conditions

You are eligible to add vehicle



Parking Management System

ADD VEHICLE

First name: dinesh

Registration No: 12111718

Email: dineshmaddi202

VEHICLE TYPE

4 WHEELER

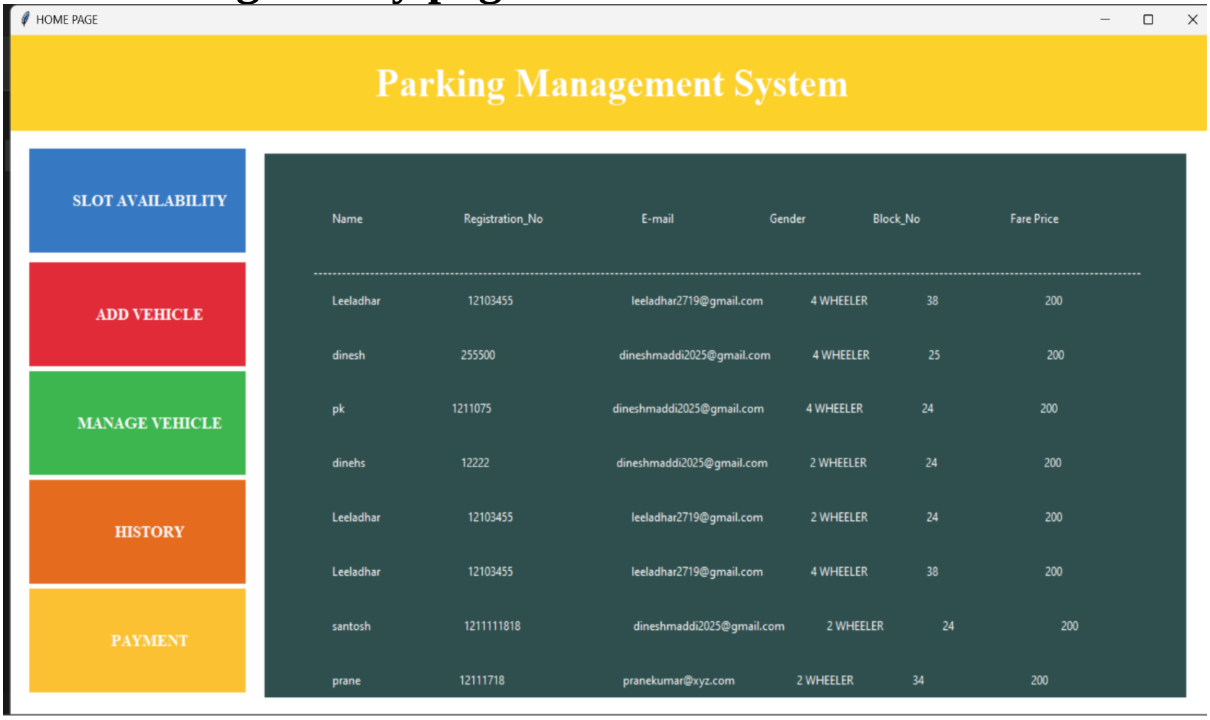
BLOCK_NUMBER: 34

☑ I Agree The Terms & Conditions

Congratulations!

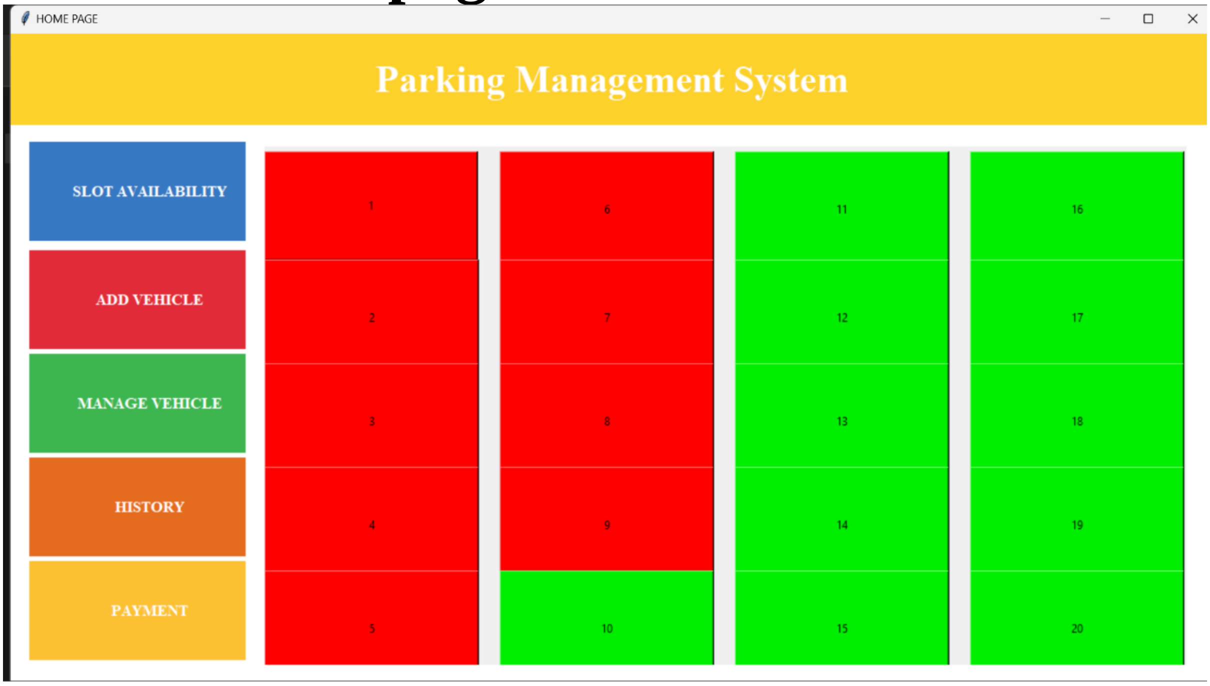
Vehicle added Successful

Slot booking history page:

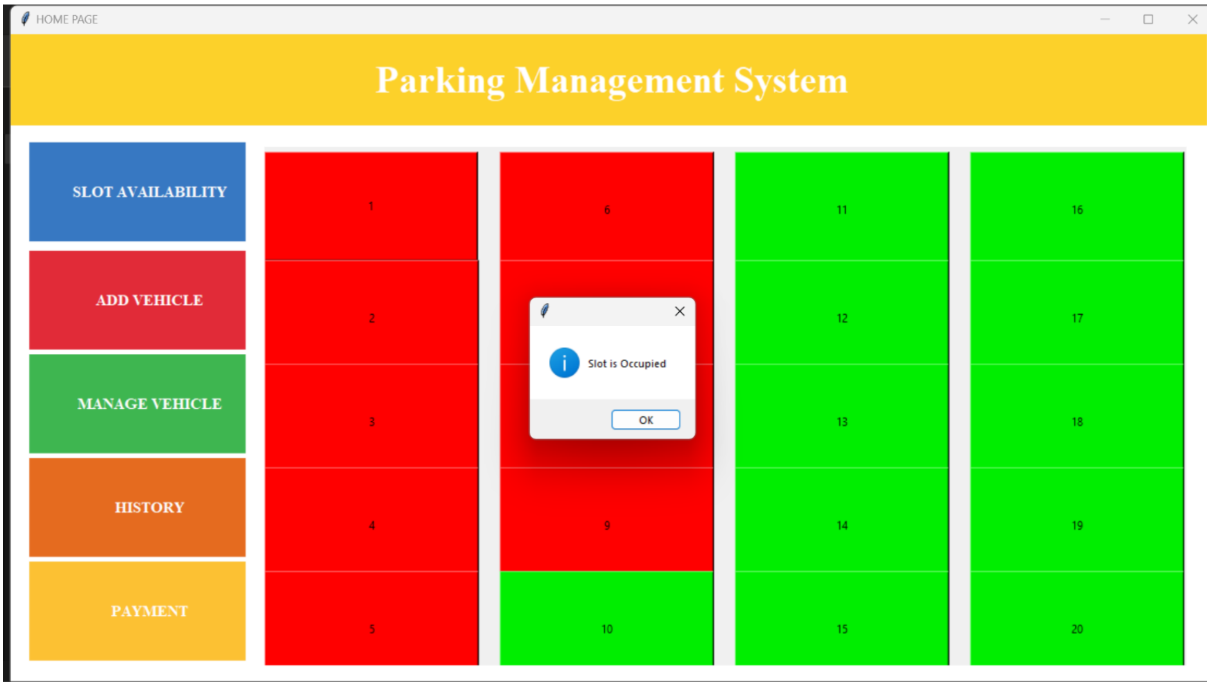


Name	Registration_No	E-mail	Gender	Block_No	Fare Price
Leeladhar	12103455	leeladhar2719@gmail.com	4 WHEELER	38	200
dinesh	255500	dineshmaddi2025@gmail.com	4 WHEELER	25	200
pk	1211075	dineshmaddi2025@gmail.com	4 WHEELER	24	200
dinehs	12222	dineshmaddi2025@gmail.com	2 WHEELER	24	200
Leeladhar	12103455	leeladhar2719@gmail.com	2 WHEELER	24	200
Leeladhar	12103455	leeladhar2719@gmail.com	4 WHEELER	38	200
santosh	1211111818	dineshmaddi2025@gmail.com	2 WHEELER	24	200
prane	12111718	pranekumar@xyz.com	2 WHEELER	34	200

Slot available page:

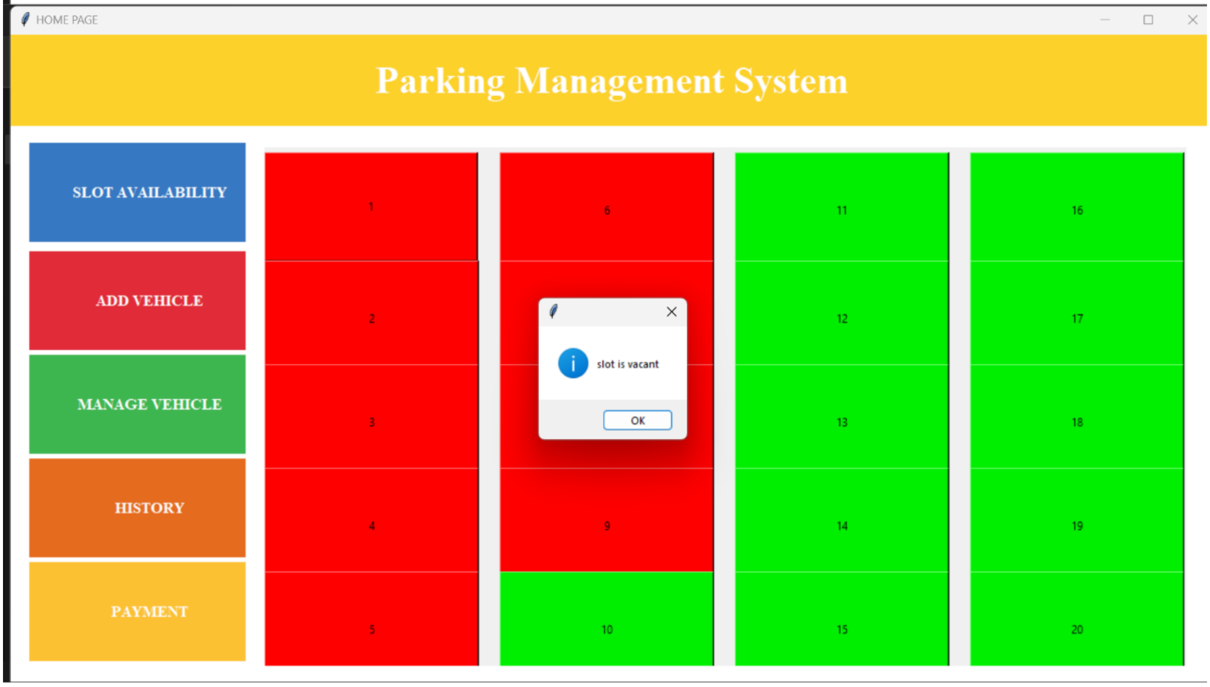


🟥 1	🟥 6	🟩 11	🟩 16
🟥 2	🟥 7	🟩 12	🟩 17
🟥 3	🟥 8	🟩 13	🟩 18
🟥 4	🟥 9	🟩 14	🟩 19
🟥 5	🟩 10	🟩 15	🟩 20



Parking Management System

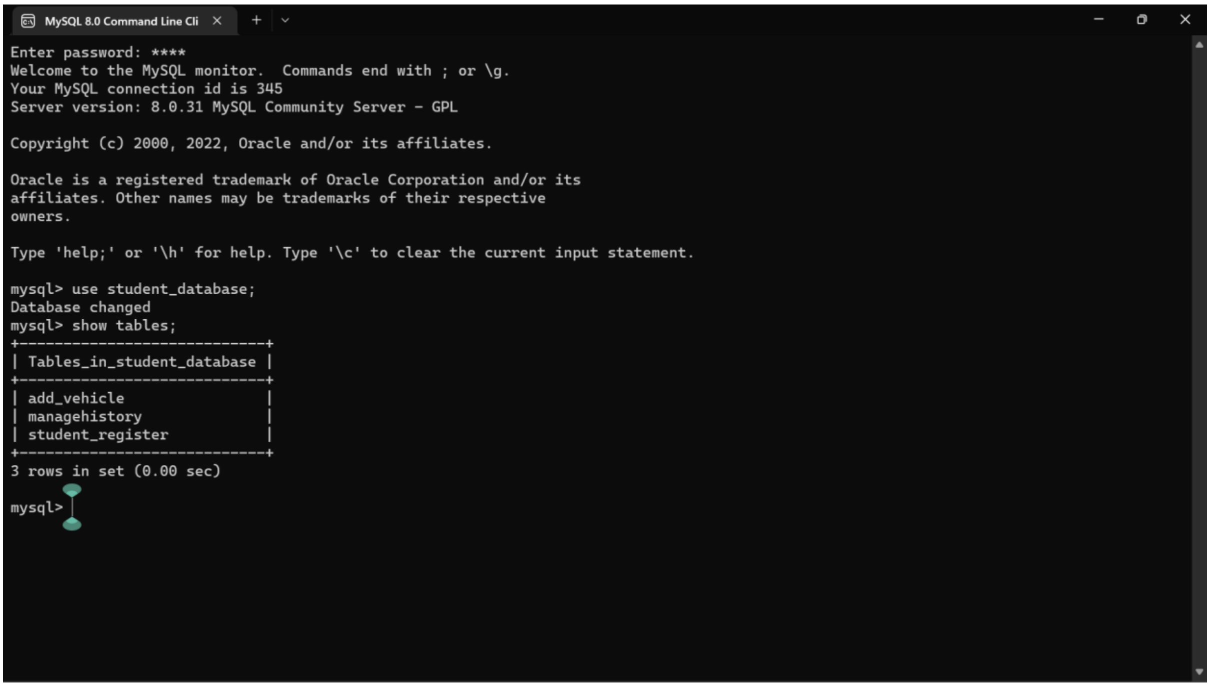
Slot is Occupied



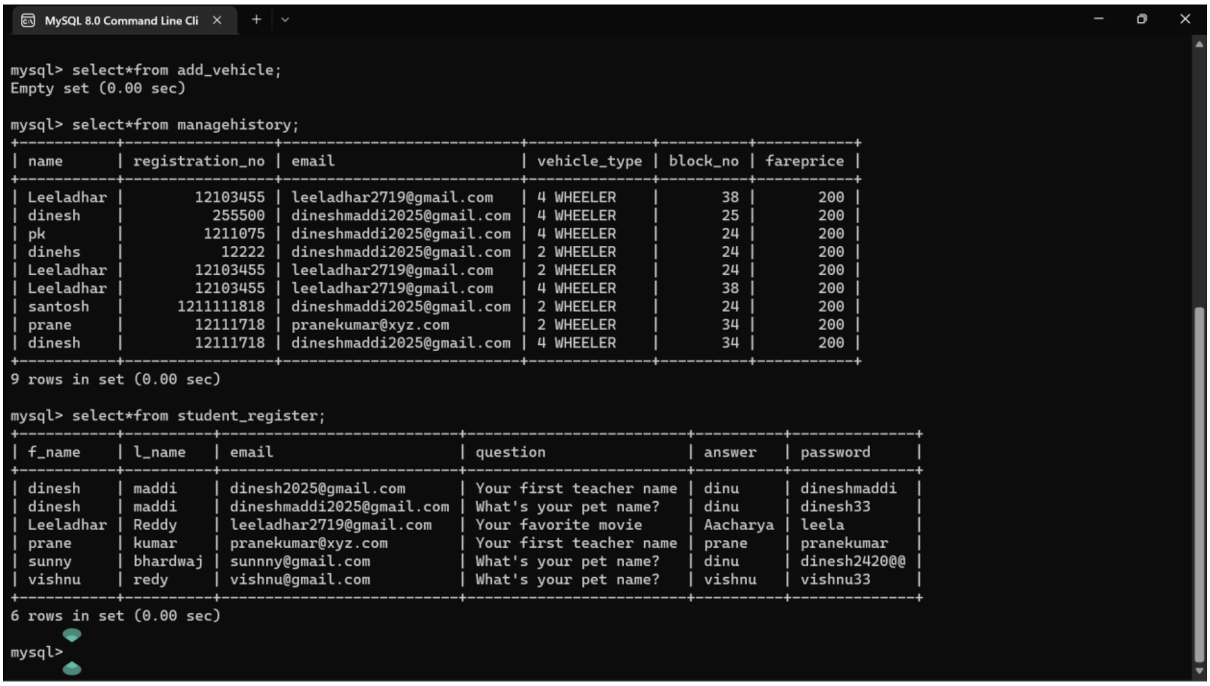
Parking Management System

slot is vacant

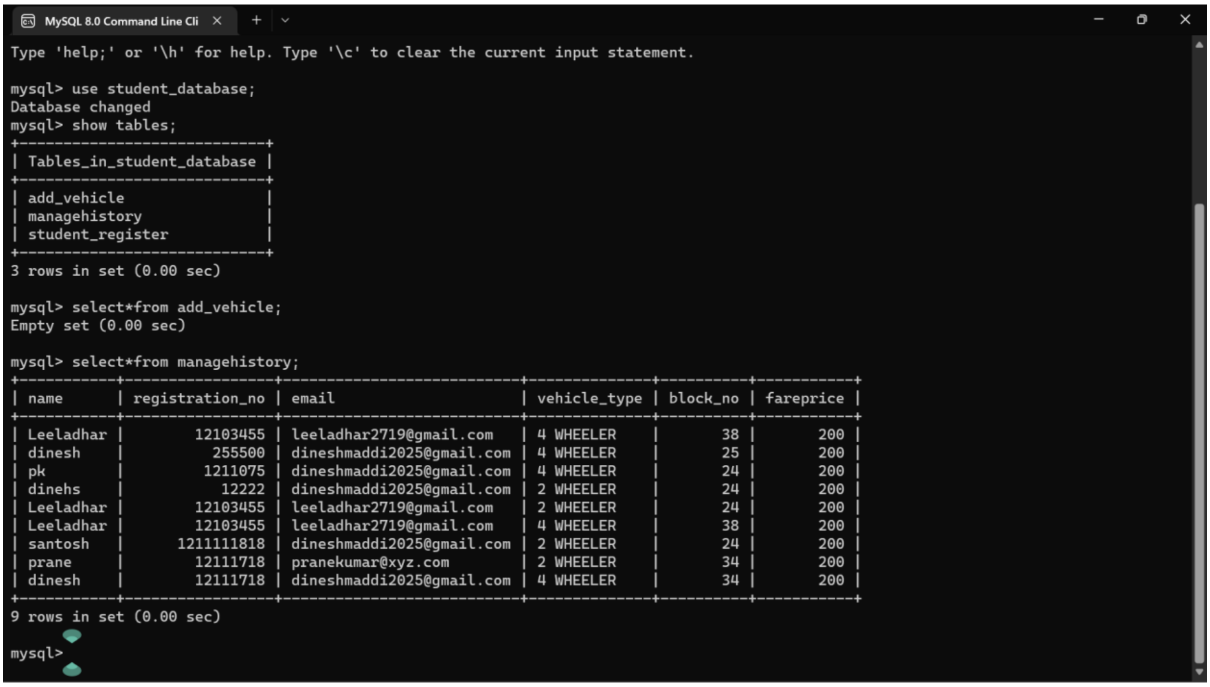
DATABASE PAGES:



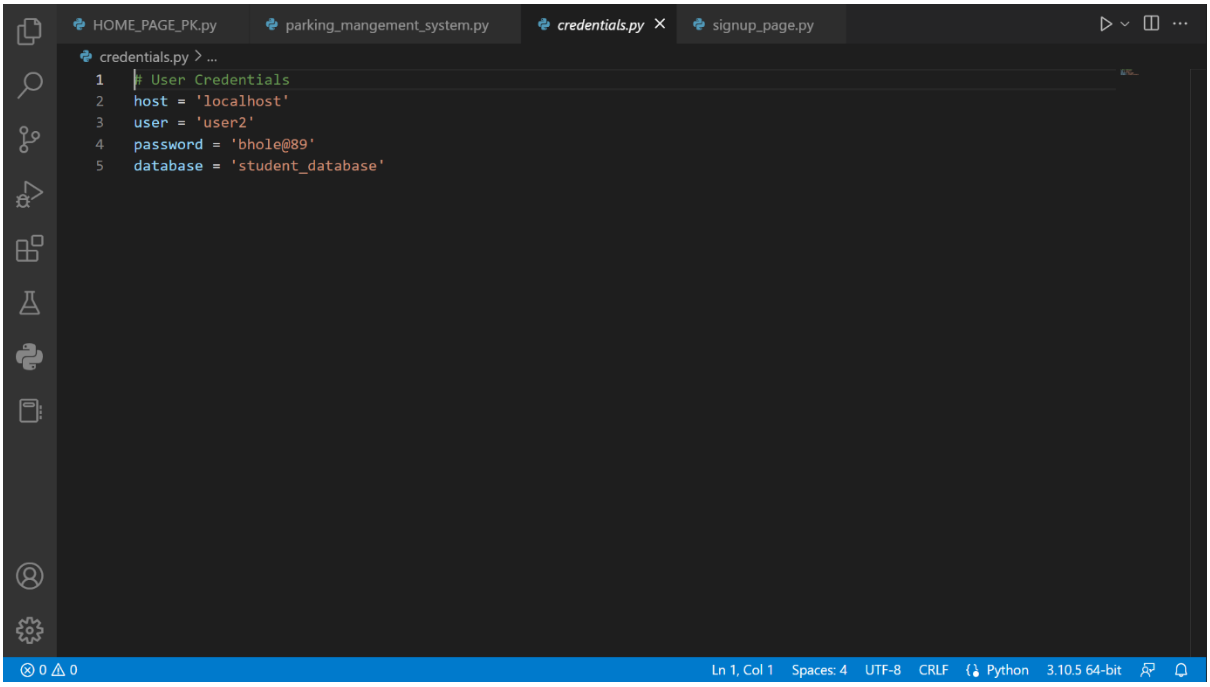
```
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 345
Server version: 8.0.31 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use student_database;
Database changed
mysql> show tables;
+----------------------------+
| Tables_in_student_database |
+----------------------------+
| add_vehicle                |
| managehistory              |
| student_register           |
+----------------------------+
3 rows in set (0.00 sec)

mysql>
```



```
mysql> select*from add_vehicle;
Empty set (0.00 sec)

mysql> select*from managehistory;
+-----------+-----------------+---------------------------+--------------+----------+-----------+
| name      | registration_no | email                     | vehicle_type | block_no | fareprice |
+-----------+-----------------+---------------------------+--------------+----------+-----------+
| Leeladhar |        12103455 | leeladhar2719@gmail.com   | 4 WHEELER    |       38 |       200 |
| dinesh    |          255500 | dineshmaddi2025@gmail.com | 4 WHEELER    |       25 |       200 |
| pk        |         1211075 | dineshmaddi2025@gmail.com | 4 WHEELER    |       24 |       200 |
| dinehs    |           12222 | dineshmaddi2025@gmail.com | 2 WHEELER    |       24 |       200 |
| Leeladhar |        12103455 | leeladhar2719@gmail.com   | 2 WHEELER    |       24 |       200 |
| Leeladhar |        12103455 | leeladhar2719@gmail.com   | 4 WHEELER    |       38 |       200 |
| santosh   |      1211111818 | dineshmaddi2025@gmail.com | 2 WHEELER    |       24 |       200 |
| prane     |        12111718 | pranekumar@xyz.com        | 2 WHEELER    |       34 |       200 |
| dinesh    |        12111718 | dineshmaddi2025@gmail.com | 4 WHEELER    |       34 |       200 |
+-----------+-----------------+---------------------------+--------------+----------+-----------+
9 rows in set (0.00 sec)

mysql> select*from student_register;
+-----------+----------+---------------------------+-------------------------+----------+--------------+
| f_name    | l_name   | email                     | question                | answer   | password     |
+-----------+----------+---------------------------+-------------------------+----------+--------------+
| dinesh    | maddi    | dinesh2025@gmail.com      | Your first teacher name | dinu     | dineshmaddi  |
| dinesh    | maddi    | dineshmaddi2025@gmail.com | What's your pet name?   | dinu     | dinesh33     |
| Leeladhar | Reddy    | leeladhar2719@gmail.com   | Your favorite movie     | Aacharya | leela        |
| prane     | kumar    | pranekumar@xyz.com        | Your first teacher name | prane    | pranekumar   |
| sunny     | bhardwaj | sunnny@gmail.com          | What's your pet name?   | dinu     | dinesh2420@@ |
| vishnu    | redy     | vishnu@gmail.com          | What's your pet name?   | vishnu   | vishnu33     |
+-----------+----------+---------------------------+-------------------------+----------+--------------+
6 rows in set (0.00 sec)

mysql>
```



```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use student_database;
Database changed
mysql> show tables;
+----------------------------+
| Tables_in_student_database |
+----------------------------+
| add_vehicle                |
| managehistory              |
| student_register           |
+----------------------------+
3 rows in set (0.00 sec)

mysql> select*from add_vehicle;
Empty set (0.00 sec)

mysql> select*from managehistory;
+-----------+-----------------+---------------------------+--------------+----------+-----------+
| name      | registration_no | email                     | vehicle_type | block_no | fareprice |
+-----------+-----------------+---------------------------+--------------+----------+-----------+
| Leeladhar |        12103455 | leeladhar2719@gmail.com   | 4 WHEELER    |       38 |       200 |
| dinesh    |          255500 | dineshmaddi2025@gmail.com | 4 WHEELER    |       25 |       200 |
| pk        |         1211075 | dineshmaddi2025@gmail.com | 4 WHEELER    |       24 |       200 |
| dinehs    |           12222 | dineshmaddi2025@gmail.com | 2 WHEELER    |       24 |       200 |
| Leeladhar |        12103455 | leeladhar2719@gmail.com   | 2 WHEELER    |       24 |       200 |
| Leeladhar |        12103455 | leeladhar2719@gmail.com   | 4 WHEELER    |       38 |       200 |
| santosh   |      1211111818 | dineshmaddi2025@gmail.com | 2 WHEELER    |       24 |       200 |
| prane     |        12111718 | pranekumar@xyz.com        | 2 WHEELER    |       34 |       200 |
| dinesh    |        12111718 | dineshmaddi2025@gmail.com | 4 WHEELER    |       34 |       200 |
+-----------+-----------------+---------------------------+--------------+----------+-----------+
9 rows in set (0.00 sec)

mysql>
```



```python
# User Credentials
host = 'localhost'
user = 'user2'
password = 'bhole@89'
database = 'student_database'
```

Codes:

THIS THE MAIN PAGE CODE WHERE TWO PAGES LINKED WITH THIS PAGE BELOW UNDERLINED PAGES ARE IMPORTED PAGES

```
from msilib.schema import ComboBox
from tkinter import *
from PIL import ImageTk
from tkinter import messagebox
from HOME_PAGE_PK import *
import credentials as cr
import pymysql
from tkinter import ttk
import subprocess
from signup_page import SignUp


class login(Frame):


    def _init_(self, root):
        self.root = root
        self.root.title("PARKING MANGEMENT SYsTEM")
        self.root.geometry("1280x800+0+0")
        self.root.resizable(True, True)
        self.lt = Label(self.root, text="Parking Management System", bg="black",
fg="white", bd=20,
                        font=("times new roman", 30, "bold"), padx=2, pady=6)

        self.lt.pack(side=TOP, fill=X)
        self.fr = Frame(self.root,  bg="black")
        self.fr.place(x=0, y=100, width=1300, height=600)
        self.bg = ImageTk.PhotoImage(file="E:/GUI USING
TKINTER/image/home.jpg")
        self.bg_image = Label(self.fr, image=self.bg).place(x=0, y=0, relheight=1,
relwidth=1)
```

```
        frame_login = Frame(self.fr, bg="white")

        frame_login.place(x=400, y=150, height=300, width=400)
        self.b1 = Button(frame_login, height=5, width=14, bg="black",
fg="white", bd=0, text="Login",
                    font=("times new roman", 13, "bold"), command=lambda:
self.Login())
        self.b2 = Button(frame_login, height=5, width=14, bg="black",
fg="white", bd=0, text="New User",
                    font=("times new roman", 13, "bold"),
command=lambda:self.redirect_window())
        self.b3 = Button(frame_login, height=5, width=14, bg="black",
fg="white", bd=0, text="MAIN MENU",
                    font=("times new roman", 13, "bold"), command=lambda:
self.Login())
        self.b1.place(x=10, y=10)
        self.b2.place(x=230, y=10)
        self.b3.place(x=130, y=170)
    def home1(self):



        self.root = root
        self.root.title("PARKING MANGEMENT SYsTEM")
        self.root.geometry("1280x800+0+0")
        self.root.resizable(False, False)
        self.lt = Label(self.root, text="Parking Management System", bg="black",
fg="white", bd=20,
                   font=("times new roman", 30, "bold"), padx=2, pady=6)

        self.lt.pack(side=TOP, fill=X)
        self.fr = Frame(self.root,  bg="black")
        self.fr.place(x=0, y=100, width=1300, height=600)
        self.bg = ImageTk.PhotoImage(file="E:/GUI USING
TKINTER/image/home.jpg")
        self.bg_image = Label(self.fr, image=self.bg).place(x=0, y=0, relheight=1,
relwidth=1)
```

```
        frame_login = Frame(self.fr, bg="white")

        frame_login.place(x=400, y=150, height=300, width=400)
        self.b1 = Button(frame_login, height=5, width=14, bg="black",
fg="white", bd=0, text="Login",
                        font=("times new roman", 13, "bold"), command=lambda:
self.Login())
        self.b2 = Button(frame_login, height=5, width=14, bg="black",
fg="white", bd=0, text="New User",
                        font=("times new roman", 13, "bold"),
command=lambda:self.redirect_window())
        self.b3 = Button(frame_login, height=5, width=14, bg="black",
fg="white", bd=0, text="MAIN MENU",
                        font=("times new roman", 13, "bold"), command=lambda:
self.Login())
        self.b1.place(x=10, y=10)
        self.b2.place(x=230, y=10)
        self.b3.place(x=130, y=170)

    def Login(self):
        self.fr = Frame(self.root, bd=10, padx=10, bg="black")
        self.fr.place(x=0, y=100, width=1300, height=600)

        self.bg = ImageTk.PhotoImage(file="E:/GUI USING
TKINTER/image/website_keyvisual_parking.jpg")
        self.bg_image = Label(self.fr, image=self.bg).place(x=0, y=0, relheight=1,
relwidth=1)
        # login frame
        fl = Frame(self.fr, bg="white")
        fl.place(x=400, y=100, height=400, width=390)
        title = Label(fl, text="Login Here", font=("Impact", 30, "bold"),
bg="white").place(x=40, y=10)
        lbl_username = Label(fl, text="Username", font=("goud old style", 15,
"bold"), bg="white", fg="gray")
        self.txt_user = Entry(fl, font=("times new roman", 15), bg="lightgray")
        lbl_username.place(x=40, y=70)
        self.txt_user.place(x=40, y=100, width=300, height=30)
        lbl_pass = Label(fl, text="password", font=("goud old style", 15, "bold"),
bg="white", fg="gray")
        lbl_pass.place(x=40, y=140)
        self.txt_pass = Entry(fl, font=("times new roman", 15), bg="lightgray")
        self.txt_pass.place(x=40, y=170, width=300, height=30)
        forget = Button(fl, text="forgot your password?", bg="white", bd=0,
fg="black",command=self.forgot_func,
                        font=("times new roman", 12)).place(x=40, y=220)
        login_btn = Button(self.root, text="Login", bg="black", fg="white",
font=("times new roman", 20),
                           command=self.login_func).place(x=460, y=480, width=140,
height=40)
        create_btn = Button(self.root, text="Create New Account",bg="black",
fg="white", font=("times new roman", 15),
                           command=self.redirect_window).place(x=620, y=480,
width=180, height=40)
        back_btn = Button(self.root, text="Back to main page",bg="black",
fg="white", font=("times new roman", 15),
                           command=self.home1).place(x=520, y=560, width=180,
height=40)

    def login_func(self):
        if self.txt_user.get() == "" and self.txt_pass.get() == "":
            messagebox.showerror("Error!", "All fields are required",
parent=self.root)
        else:
            try:
                connection = pymysql.connect(host=cr.host, user=cr.user,
password=cr.password, database=cr.database)
                cur = connection.cursor()
                cur.execute("select * from student_register where email=%s and
password=%s",
                            (self.txt_user.get(), self.txt_pass.get()))
                row = cur.fetchone()
                if row == None:
                    messagebox.showerror("Error!", "Invalid USERNAME &
PASSWORD", parent=self.root)

                else:
                    messagebox.showinfo("Success","Wellcome to PARKING
MANAGEMENT SYSTEM",parent=self.root)
                    # Clear all the entries
                    self.redirect_window1()
                    self.reset_fields()

                    connection.close()
            except Exception as e:
                messagebox.showerror("Error!", f"Error due to {str(e)}",
parent=self.root)



    def forgot_func(self):
        if self.txt_user.get() == "":
            messagebox.showerror("Error!", "Please enter your User Id",
parent=self.root)
        else:
            try:
                connection = pymysql.connect(host=cr.host, user=cr.user,
password=cr.password, database=cr.database)
                cur = connection.cursor()
                cur.execute("select * from student_register where email=%s",
self.txt_user.get())
                row = cur.fetchone()
                if row == None:
                    messagebox.showerror("Error!", "userid doesn't exists")
                else:
                    connection.close()

                    # =========================SECOND
WINDOW=============================
                    # ------------Toplevel:create a window top of another window--------
---
                    # ------------focus_force:Helps to to focus on the current window----
--
                    # -----Grab:Helps to grab the current window until user ungrab it---
---

                    self.root = Toplevel()
                    self.root.title("Forget Password?")
                    self.root.geometry("400x440+450+200")
                    self.root.config(bg="white")
                    self.root.focus_force()
                    self.root.grab_set()
                    title3 = Label(self.root, text="Change your password", font=("times
new roman", 20, "bold"),
                      bg="white").place(x=10, y=10)

                    title4 = Label(self.root, text="It's quick and easy", font=("times
new roman", 12),
                      bg="white").place(x=10, y=45)

                    title5 = Label(self.root, text="Select your question", font=("times
new roman", 15, "bold"),
                      bg="white").place(x=10, y=85)

                    self.sec_ques = ttk.Combobox(self.root, font=("times new roman",
13), state='readonly',
                                justify=CENTER)
                    self.sec_ques['values'] = (
                    "Select", "What's your pet name?", "Your first teacher name",
"Your birthplace",
                    "Your favorite movie")
                    self.sec_ques.place(x=10, y=120, width=270)
                    self.sec_ques.current(0)

                    title6 = Label(self.root, text="Answer", font=("times new roman",
15, "bold"), bg="white").place(
            x=10, y=160)

                    self.ans = Entry(self.root, font=("arial"))
                    self.ans.place(x=10, y=195, width=270)

                    title7 = Label(self.root, text="New Password", font=("times new
roman", 15, "bold"),
                      bg="white").place(x=10, y=235)

                    self.new_pass = Entry(self.root, font=("arial"))
                    self.new_pass.place(x=10, y=270, width=270)

                    self.create_button = Button(self.root, text="Submit",
command=self.change_pass,
                                  font=("times new roman", 18, "bold"), bd=0,
cursor="hand2", bg="green2",
                                  fg="white").place(x=95, y=340, width=200)
```

```
                    #
================================================================
================

            except Exception as e:
                messagebox.showerror("Error", f"{e}")

    def change_pass(self):
        if self.txt_user.get() == "" or self.sec_ques.get() == "Select" or
self.new_pass.get() == "":
            messagebox.showerror("Error!", "Please fill the all entry field
correctly")
        else:
            try:
                connection = pymysql.connect(host=cr.host, user=cr.user,
password=cr.password, database=cr.database)
                cur = connection.cursor()
                cur.execute("select * from student_register where email=%s and
question=%s and answer=%s",
                            (self.txt_user.get(), self.sec_ques.get(), self.ans.get()))
                row = cur.fetchone()

                if row == None:
                    messagebox.showerror("Error!", "Please fill the all entry field
correctly")
                else:
                    try:
                        cur.execute("update student_register set password=%s where
email=%s",
                                    (self.new_pass.get(), self.txt_user.get()))
                        connection.commit()

                        messagebox.showinfo("Successful", "Password has changed
successfully")
                        connection.close()
                        self.reset_fields()
                        self.root.destroy()

                    except Exception as er:
                        messagebox.showerror("Error!", f"{er}")

            except Exception as er:
                messagebox.showerror("Error!", f"{er}")
    def redirect_window(self):
         self.root.destroy()
         # Importing the signup window.
        # The page must be in the same directory
         root = Tk()
         obj = SignUp(root)
         root.mainloop()
    def redirect_window1(self):
         self.root.destroy()
         # Importing the signup window.
        # The page must be in the same directory
         root = Tk()
         obj = home(root)
         root.mainloop()
    def reset_fields(self):
        self.txt_user.delete(0, END)
        self.txt_pass.delete(0, END)
if _name_ == "_main_":
     root = Tk()
     obj = login(root)
     root.mainloop()
```

THIS IS HOME PAGE CODE

```
import configparser
from msilib.schema import ComboBox
from tkinter import *
from tkinter import messagebox, ttk

import mysql.connector
import pymysql
from PIL import Image, ImageTk

import credentials as cr


class home:
    def __init__(self, root):
        self.root = root
        self.root.title("HOME PAGE")
        self.root.geometry("1280x800+0+0")
        self.root.config(bg="white")
        self.lt = Label(self.root, text="Parking Management System", bg="#FCD12A",
fg="white", bd=20,
                        font=("times new roman", 30, "bold"), padx=2, pady=6)
        self.lt.pack(side=TOP, fill=X)
        self.fr = Frame(self.root, bd=10, padx=10, bg="white")
        self.fr.place(x=0, y=100, width=250, height=600)
        self.b1 = Button(self.fr, height=5, width=25, bg="#3778C2", fg="white", bd=0,
text="HOME",
                         font=("times new roman", 13, "bold"), command=lambda: self.home_fun())
        self.b2 = Button(self.fr, height=5, width=25, bg="#E12B38", fg="white", bd=0,
text="ADD VEHICLE",
                         font=("times new roman", 13, "bold"), command=lambda: self.vecy_fun())
        self.b3 = Button(self.fr, height=5, width=25, bg="#3EB650", fg="white", bd=0,
text="MANAGE VEHICLE",
                         font=("times new roman", 13, "bold"), command=lambda: self.macy_fun())
        self.b4 = Button(self.fr, height=5, width=25, bg="#E56B1F", fg="white", bd=0,
text="HISTORY",
                         font=("times new roman", 13, "bold"), command=lambda: self.hist_fun())
        self.b5 = Button(self.fr, height=5, width=25, bg="#FCC133", fg="white", bd=0,
text="PAYMENT",
                         font=("times new roman", 13, "bold"), command=lambda:
self.payment_fun())
        self.b1.place(x=0, y=5)
        self.b2.place(x=0, y=120)
        self.b3.place(x=0, y=230)
        self.b4.place(x=0, y=340)
        self.b5.place(x=0, y=450)
        self.fr = Frame(self.root, bg="black")
        self.fr.place(x=270, y=120, width=980, height=550)

        self.bg = ImageTk.PhotoImage(file="E:/GUI USING TKINTER/image/Parking-
Management-System1.jpg")
        self.bg_image = Label(self.fr, image=self.bg).place(x=0, y=0, relheight=1, relwidth=1)

    def home_fun(self):
        connection = pymysql.connect(host=cr.host, user=cr.user, password=cr.password,
database=cr.database)
        cur = connection.cursor()
        cur.execute("select count(*) from managehistory;")
        global t1
        global slot
        slot=[]
        for i in cur:
            t1=i
        fr1 = Frame(self.root)
        fr1.place(x=270, y=120, width=980, height=550)
        b1 = Button(fr1, text="1", bg="green", height=7,command=lambda: self.myclick1(),
width=31)
        b1.place(x=0, y=5)
        b2 = Button(fr1, text="2", padx=40, pady=20, bg="green", height=5, width=20)
        b2.place(x=0, y=120)
        b3 = Button(fr1, text="3", padx=40, pady=20, height=5, width=20, bg="green")
        b3.place(x=0, y=230)
        b4 = Button(fr1, text="4", padx=40, pady=20, command=lambda: self.myclick1(),
height=5, width=20, bg="green2")
        b4.place(x=0, y=340)
        b5 = Button(fr1, text="5", padx=40, pady=20, command=lambda: self.myclick1(),
height=5, width=20, bg="green2")
        b5.place(x=0, y=450)
        b6 = Button(fr1, text="6", padx=40, pady=20, command=lambda: self.myclick1(),
height=5, width=20, bg="green2")
        b6.place(
            x=250, y=5)
        b7 = Button(fr1, text="7", padx=40, pady=20, command=lambda: self.myclick1(),
height=5, width=20, bg="green2")
        b7.place(
            x=250, y=120)
        b8 = Button(fr1, text="8", padx=40, pady=20, command=lambda: self.myclick1(),
height=5, width=20, bg="green2")
        b8.place(
            x=250, y=230)
        b9 = Button(fr1, text="9", padx=40, pady=20, command=lambda: self.myclick1(),
height=5, width=20, bg="green2")
        b9.place(
            x=250, y=340)
        b10 = Button(fr1, text="10", padx=40, pady=20, command=lambda: self.myclick1(),
height=5, width=20, bg="green2")
        b10.place(
            x=250, y=450)
        b11 = Button(fr1, text="11", padx=40, pady=20, command=lambda: self.myclick1(),
height=5, width=20, bg="green2")
        b11.place(
            x=500, y=5)
        b12 = Button(fr1, text="12", padx=40, pady=20, command=lambda: self.myclick1(),
height=5, width=20, bg="green2")
        b12.place(
            x=500, y=120)
        b13 = Button(fr1, text="13", padx=40, pady=20, command=lambda: self.myclick1(),
height=5, width=20, bg="green2")
        b13.place(
            x=500, y=230)
        b14 = Button(fr1, text="14", padx=40, pady=20, command=lambda: self.myclick1(),
height=5, width=20, bg="green2")
        b14.place(
            x=500, y=340)
        b15 = Button(fr1, text="15", padx=40, pady=20, command=lambda: self.myclick1(),
height=5, width=20, bg="green2").place(
            x=500, y=450)
        b16 = Button(fr1, text="16", padx=40, pady=20, command=lambda: self.myclick1(),
height=5, width=20, bg="green2").place(
            x=750, y=5)
        b17 = Button(fr1, text="17", padx=40, pady=20, command=lambda: self.myclick1(),
height=5, width=20, bg="green2").place(
            x=750, y=120)
        b18= Button(fr1, text="18", padx=40, pady=20, command=lambda: self.myclick1(),
height=5, width=20, bg="green2").place(
            x=750, y=230)
        b19 = Button(fr1, text="19", padx=40, pady=20, command=lambda: self.myclick1(),
height=5, width=20, bg="green2").place(
            x=750, y=340)
        b20 = Button(fr1, text="20", padx=40, pady=20, command=lambda: self.myclick1(),
height=5, width=20, bg="green2").place(
            x=750, y=450)
        ba = [b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14,b15,b16,b17,b18,b19,b20]
        for i in range(t1[0]):
            ba[i].config(bg="red")
            ba[i].config(command=lambda: self.myclick0())




    def vecy_fun(self):
        fr1 = Frame(self.root)
        fr1.place(x=270, y=120, width=980, height=550)
        fr2 = Frame(fr1, bg="white")
        fr2.place(x=100, y=0, width=500, height=550)
        global fname_txt
        global lname_txt
        global email_txt
        global questions
        global XYZ
        title1 = Label(fr2, text="ADD VEHICLE", font=("times new roman", 25, "bold"),
bg="white").place(x=20, y=10)

        f_name = Label(fr2, text="First name", font=("helvetica", 15, "bold"),
bg="white").place(x=20, y=100)
        l_name = Label(fr2, text="Registration No", font=("helvetica", 15, "bold"),
bg="white").place(x=240, y=100)

        fname_txt = Entry(fr2, font=("arial"))
        fname_txt.place(x=20, y=130, width=200)

        lname_txt = Entry(fr2, font=("arial"))
        lname_txt.place(x=240, y=130, width=200)

        email = Label(fr2, text="Email", font=("helvetica", 15, "bold"),
bg="white").place(x=20, y=180)

        email_txt = Entry(fr2, font=("arial"))
        email_txt.place(x=20, y=210, width=420)
        sec_question = Label(fr2, text="VEHICLE TYPE", font=("helvetica", 15, "bold"),
bg="white").place(x=20, y=260)
        XY1= Label(fr2, text="BLOCK_NUMBER", font=("helvetica", 15, "bold"),
bg="white").place(x=20, y=320)
        XYZ = Entry(fr2, font=("arial"))
        XYZ.place(x=20, y=350, width=420)

        questions = ttk.Combobox(fr2, font=("helvetica", 13), state='readonly',
justify=CENTER)
        questions['values'] = ("Select", "2 WHEELER", "4 WHEELER")
        questions.place(x=120, y=300, width=200)
        questions.current(0)

        self.terms = IntVar()
        terms_and_con = Checkbutton(fr2, text="I Agree The Terms & Conditions",
variable=self.terms, onvalue=1,
                                    offvalue=0, bg="white", font=("times new roman", 12)).place(x=20,
y=380)
        self.signup = Button(fr2, text="ADD VEHICLE", font=("times new roman", 18,
"bold"), bd=0, cursor="hand2",
                             bg="green2", fg="white", command=self.login_func)
        self.signup.place(x=50, y=430, width=250)

    def login_func(self):
        if fname_txt.get() == "" or lname_txt.get() == "" or email_txt.get() == "" or
questions.get() == "Select":
            messagebox.showerror("Error!", "Sorry!, All fields are required", parent=self.root)

        elif self.terms.get() == 0:
            messagebox.showerror("Error!", "Please Agree with our Terms & Conditions",
parent=self.root)

        else:
            connection = pymysql.connect(host=cr.host, user=cr.user, password=cr.password,
database=cr.database)
            cur = connection.cursor()
            cur.execute("select email from student_register where email=%s", email_txt.get())
            row = cur.fetchone()

            # Check if th entered email id is already exists or not.
            if row != None:
                messagebox.showinfo("", "You are eligible to add vehicle", parent=self.root)

                sql = "insert into managehistory values(%s,%s,%s,%s,%s,%s);"
                sql1 = [fname_txt.get(), lname_txt.get(), email_txt.get(),
questions.get(),XYZ.get(),200]
                cur.execute(sql, sql1)
                connection.commit()
                connection.close()
                messagebox.showinfo("Congratulations!", "Vehicle added Successful",
parent=self.root)
                self.reset_fields()

    def macy_fun(self):
        labelFrame = Frame(self.root, bg='dark slate grey')
        labelFrame.place(x=270, y=120, height=550, width=980)
        y = 0.25
        Label(labelFrame, text="%-40s%-50s%-40s%-30s%-40s%-40s" % (
        'Name', 'Registration_No', 'E-mail', 'Gender', 'Block_No', 'Fare Price'), bg='dark slate
grey',
              fg='white').place(relx=0.07, rely=0.1)
        Label(labelFrame,
              text="----------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------",
              bg='dark slate grey', fg='white').place(relx=0.05, rely=0.2)
        connection = pymysql.connect(host=cr.host, user=cr.user, password=cr.password,
database=cr.database)
        cur = connection.cursor()
        sql = "select * from managehistory;"
        cur.execute(sql)
        connection.commit()
        for i in cur:
            Label(labelFrame, text="%-40s%-50s%-40s%-30s%-40s%-40s" % (i[0], i[1], i[2],
i[3], i[4], i[5]),
                    bg='dark slate grey', fg='white').place(relx=0.07, rely=y)
            y += 0.1
        return

    def hist_fun(self):
        return


    def myclick0(self):
        messagebox.showinfo("","Slot is Occupied")
    def payment_fun():
        return
    def myclick1(self):
        messagebox.showinfo("","slot is vacant")



if __name__ == "__main__":
    root = Tk()
    obj = home(root)
    root.mainloop()
```

Signup page code:

```
from tkinter import *
from PIL import Image, ImageTk
from tkinter import ttk, messagebox
import pymysql, os
import credentials as cr
from HOME_PAGE_PK import *




class SignUp:
    def _init__(self, root):
        self.window = root
        self.window.title("Sign Up")
        self.window.geometry("1280x800+0+0")
        self.window.config(bg = "white")

        self.bg_img = ImageTk.PhotoImage(file="E:/GUI USING
TKINTER/image/website_keyvisual_parking.jpg")
        background =
Label(self.window,image=self.bg_img).place(x=0,y=0,relwidth=1,relheight=1)


        frame = Frame(self.window, bg="white")
        frame.place(x=350,y=100,width=500,height=550)

        title1 = Label(frame, text="Sign Up", font=("times new
roman",25,"bold"),bg="white").place(x=20, y=10)
        title2 = Label(frame, text="Join with us", font=("times new roman",13),bg="white",
fg="gray").place(x=20, y=50)

        f_name = Label(frame, text="First name",
font=("helvetica",15,"bold"),bg="white").place(x=20, y=100)
        l_name = Label(frame, text="Last name",
font=("helvetica",15,"bold"),bg="white").place(x=240, y=100)

        self.fname_txt = Entry(frame,font=("arial"))
        self.fname_txt.place(x=20, y=130, width=200)

        self.lname_txt = Entry(frame,font=("arial"))
        self.lname_txt.place(x=240, y=130, width=200)

        email = Label(frame, text="Email",
font=("helvetica",15,"bold"),bg="white").place(x=20, y=180)

        self.email_txt = Entry(frame,font=("arial"))
        self.email_txt.place(x=20, y=210, width=420)
        sec_question = Label(frame, text="Security questions",
font=("helvetica",15,"bold"),bg="white").place(x=20, y=260)
        answer = Label(frame, text="Answer",
font=("helvetica",15,"bold"),bg="white").place(x=240, y=260)

        self.questions =
ttk.Combobox(frame,font=("helvetica",13),state='readonly',justify=CENTER)
        self.questions['values'] = ("Select","What's your pet name?","Your first teacher
name","Your birthplace", "Your favorite movie")
        self.questions.place(x=20,y=290,width=200)
        self.questions.current(0)

        self.answer_txt = Entry(frame,font=("arial"))
        self.answer_txt.place(x=240, y=290, width=200)

        password =  Label(frame, text="New password",
font=("helvetica",15,"bold"),bg="white").place(x=20, y=340)

        self.password_txt = Entry(frame,font=("arial"))
        self.password_txt.place(x=20, y=370, width=420)

        self.terms = IntVar()
        terms_and_con = Checkbutton(frame,text="I Agree The Terms &
Conditions",variable=self.terms,onvalue=1,offvalue=0,bg="white",font=("times new
roman",12)).place(x=20,y=420)
        self.signup = Button(frame,text="Sign Up",command=self.signup_func,font=("times
new roman",18,
"bold"),bd=0,cursor="hand2",bg="green2",fg="white").place(x=10,y=470,width=250)
        self.signup = Button(frame,text="Back to main page",font=("times new roman",18,
"bold"),bd=0,cursor="hand2",bg="green2",fg="white").place(x=240,y=470,width=250)


    def signup_func(self):
        if self.fname_txt.get()=="" or self.lname_txt.get()=="" or self.email_txt.get()=="" or
self.questions.get()=="Select" or self.answer_txt.get()=="" or self.password_txt.get() == "":
            messagebox.showerror("Error!","Sorry!, All fields are required",parent=self.window)

        elif self.terms.get() == 0:
            messagebox.showerror("Error!","Please Agree with our Terms &
Conditions",parent=self.window)

        else:
            try:
                connection = pymysql.connect(host=cr.host, user=cr.user, password=cr.password,
database=cr.database)
                cur = connection.cursor()
                cur.execute("select * from student_register where email=%s",self.email_txt.get())
                row=cur.fetchone()

                # Check if th entered email id is already exists or not.
                if row!=None:
                    messagebox.showerror("Error!","The email id is already exists, please try again
with another email id",parent=self.window)
                else:
                    cur.execute("insert into student_register
(f_name,l_name,email,question,answer,password) values(%s,%s,%s,%s,%s,%s)",
                                    (
                                        self.fname_txt.get(),
                                        self.lname_txt.get(),
                                        self.email_txt.get(),
                                        self.questions.get(),
                                        self.answer_txt.get(),
                                        self.password_txt.get()
                                    ))
                    connection.commit()
                    connection.close()
                    messagebox.showinfo("Congratulations!","Register
Successful",parent=self.window)
                    self.reset_fields()
            except Exception as es:
                messagebox.showerror("Error!",f"Error due to {es}",parent=self.window)



    def reset_fields(self):
        self.fname_txt.delete(0, END)
        self.lname_txt.delete(0, END)
        self.email_txt.delete(0, END)
        self.questions.current(0)
        self.answer_txt.delete(0, END)
        self.password_txt.delete(0, END)

if __name_ == "__main__":
    root = Tk()
    obj = SignUp(root)
    root.mainloop()
```

Conclusion

This study was conducted by the researchers to develop and implement a Vehicle Parking Management System using Python. The developed system was presented to the target users and respondents for assessment. The result of the assessment showed that the developed system is an effective tool to increase the efficiency and services offered in vehicle parking areas. The developed system can indeed provide efficient and convenient parking services to the customers.