CHINMAYA VIDYALAYA
SENIOR SECONDARY SCHOOL

Thondamuthur Road, Vadavalli, Coimbatore-641046

AISSCE 2023-2024

COMPUTER SCIENCE

PROJECT

CAR MARKETPLACE APPLICATION

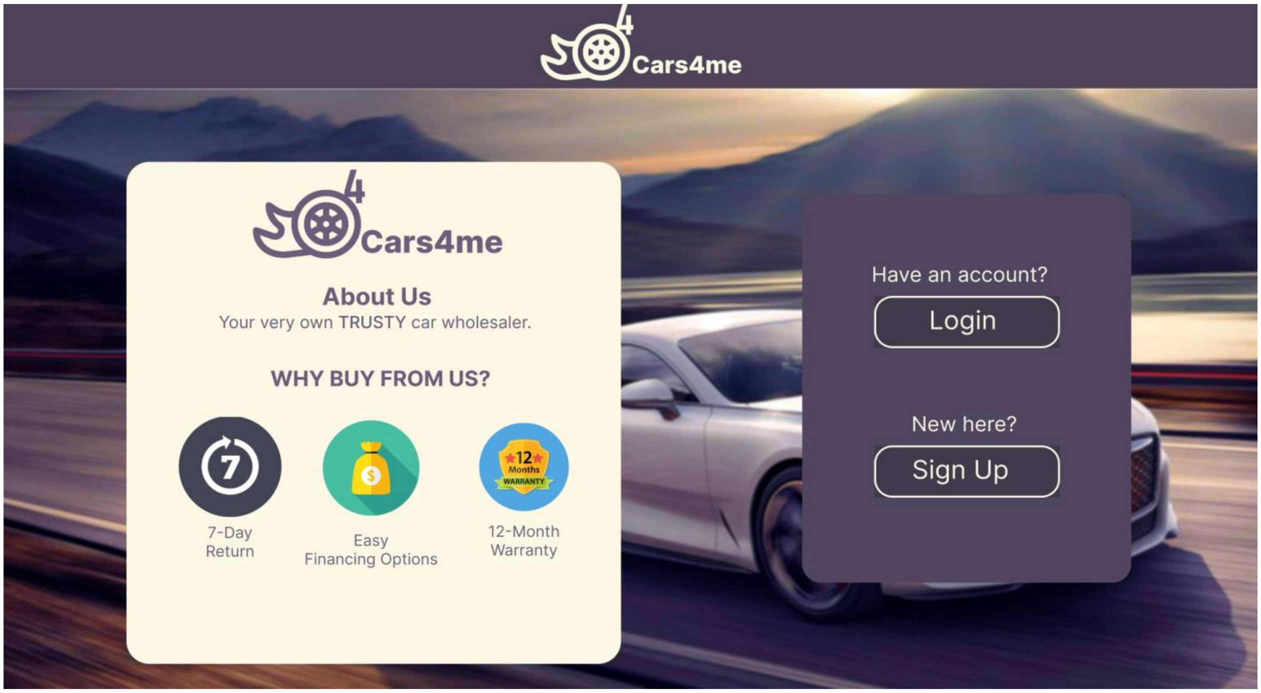
SUBMITTED BY

JULIUS B. THOMAS – XII A

EXAM NO:

CAR MARKETPLACE APPLICATION

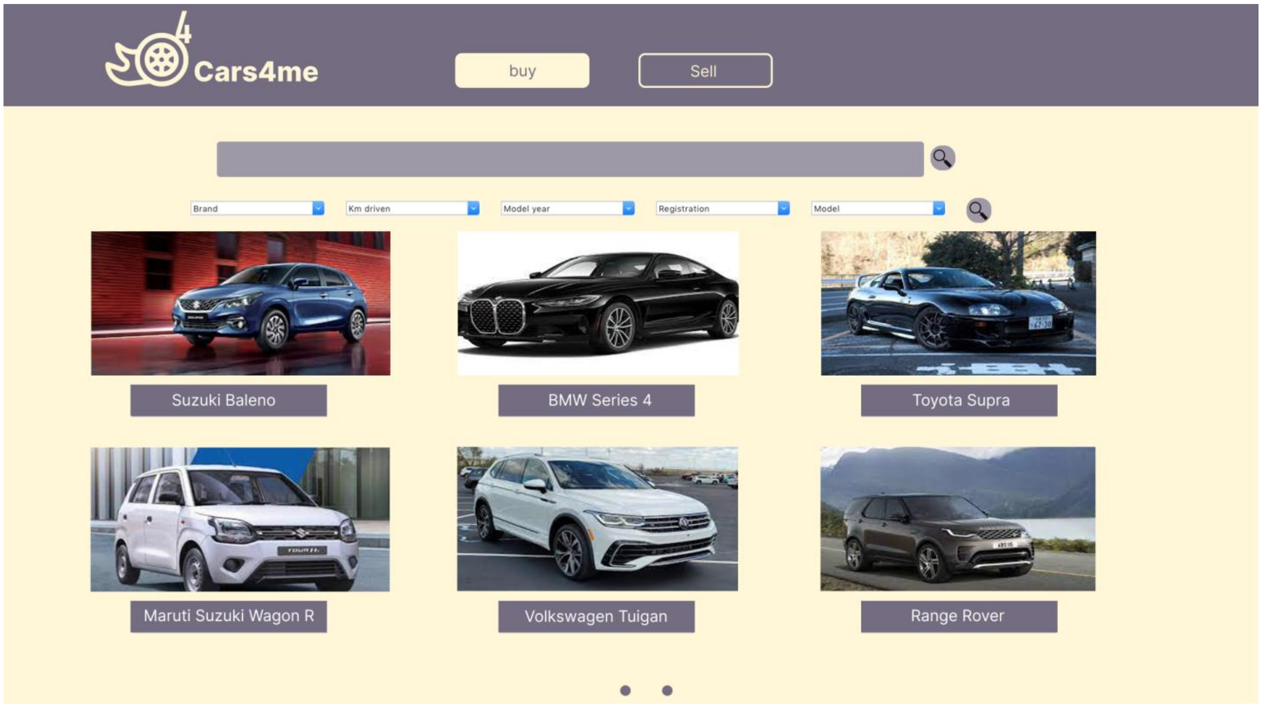
(USING PYTHON AND MySQL)



About Us

Your very own TRUSTY car wholesaler.

WHY BUY FROM US?



ACKNOWLEDGEMENT

Upon the successful completion of the project, I would like to express my sincere thanks and my heartfelt obligation towards all those who have been of help throughout the journey of this project. Their impeccable guidance and continuous encouragement aided me to complete the task in the allocated period of time.

I would also like to extend my unfeigned gratitude and my utmost gratefulness to Ms. Neelaveni D for giving me a remarkable opportunity to explore and learn but most importantly reflect. Her valuable support and able guidance were stepping stones towards the success of the project , and if it were not for her continual support, this would not have been possible.

I am extremely thankful to my principal Ms. Subhashini Ramakrishnan for the moral support extended during the tenure of the project. I am truly grateful to the kind patronage, inspirations, timely guidance, encouragement and suggestions, which immensely contributed to the evolution of my ideas on this project. Lastly, I would like to thank my family and friends for their unending support and faith in me.

DATE NAME OF THE STUDENT

TABLE OF CONTENTS

SYNOPSIS

SYNOPSIS

The Cars4me application is a cutting-edge digital platform that revolutionizes the process of buying and selling vehicles. Designed with user convenience in mind, the application offers a multi-faceted approach to streamline the entire automotive transaction journey.

For sellers, a dedicated selling page simplifies the process of listing vehicles. With an intuitive interface, sellers can input detailed information about their cars, upload high-quality images, and engage in secure communication with potential buyers.

When users find a car of interest, the viewing page offers an immersive experience. High-resolution images, detailed specifications, and seller-provided descriptions provide a comprehensive understanding of the vehicle.

This program has been designed using Python 3.7.1 (64 bits) as the frontend. VS CODE was used as source code editor. It is a very efficient and easy to use editor as it supports many languages and has tools for debugging, syntax highlighting etc. The tkinter module was used for designing the framework of this system.

This program uses MySQL as a Back-end to store data using relations. MySQL is an Oracle-backed open-source relational database management system (RDBMS) based on Structured Query Language (SQL).Although it can be used in a wide range of applications, MySQL is not often associated with web applications and online publishing.

INTRODUCTION ABOUT THE PROJECT

INTRODUCTION ABOUT THE PROJECT

The ”Cars4me” allows efficient administration and effective monitoring of the users records and data in a user-friendly manner.

SIGN-IN WINDOW:

The login window opens right after clicking the sign in button on the opening window. The user should register by clicking the sign up button and enter the username and password. Then clicking the sign in button and entering the right credentials the user can access the features. User can click sign up button if user does not have an account

SIGN-UP WINDOW:

The login window opens right after clicking the sign up button on the opening window. The user should register by entering new username and password. Then clicking the sign up button and entering the right credentials the user can access the features. User can click sign in button if user does have an account

HOME WINDOW / LISTING WINDOW:

The different windows of the application are accessed via the home window.

The home window has 3 main sections namely:

- ADD CARS
- BUY CARS
- VIEW CARS

SELL CARS:

The car-selling page within our application offers an effortless and user-friendly platform for individuals looking to sell their vehicles. Streamlined and intuitive, this page guides sellers through the process step by step, ensuring they can easily list their cars for prospective buyers to discover.

BUY CARS:

The car-buying page within our application offers a seamless and convenient experience for individuals in search of their ideal vehicle. Designed with user needs in mind, this page simplifies the process of finding and purchasing cars, making it an enjoyable journey from exploration to ownership.

VIEW CARS:

The car-viewing page offers a comprehensive and immersive experience, allowing users to delve deeply into the details of their selected vehicle. Designed to provide a closer look, this page offers an array of features that help users make well-informed decisions before finalizing a purchase.

SYSTEM CONFIGURATION

SYSTEM CONFIGURATION

SOFTWARE USED:

Pyhton 3.7.1 (64 bits)	- Front End
Visual Studio Code	- Source Code Editor
macOS High Sierra 10.13.6	- Operating System
MySQL 5.7Community Server	- Back End

HARDWARE USED:

Hard Disk Drive Capacity	- 512 GB
Processor ,	- 2.5 GHz Intel Core i5
RAM	- 12 GB
Output Device	- 1920x1080 Monitor (resolution)
Input Device	- Mouse, keyboard

ABOUT PYTHON

ABOUT PYTHON

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was created in the late 1980s, and first released in 1991 by Guido Van Rossum. It is based on the languages ABC and Modula 3.

ADVANTAGES:

- Versatile, easy to use as beginner programmers can learn the easy syntax quickly.
- Interpreted language, which compiles line by line during runtime and presents errors with a trace back.
- Open source with a vibrant community.
- Free learning sources are available throughout the Internet.

DISADVANTAGES:

- Python has high memory consumption due to being interpreted.
- Python is slow at runtime, especially during initialization of a program, due to its interpreted architecture.
- It has much lesser libraries than other languages like C, C++
- It is difficult to convert to other languages as it is dynamically typed.

MODULE USED

MODULES USED

Tkinter:

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications.

PIL:

Python Imaging Library is a free and open-source additional library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats.

Datetime:

Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an objectin Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.

Mysql.connector:

mysql.connector is a module that enables Python programs to access MySQL databases, using an API that is compliant with the Python Database API Specification v2.0 (PEP 249). It is written in pure Python and does not have any dependencies except for the Python Standard Library. There are various versions of MySQL Connector/Python available.

io:

In Python, io is a built-in module that provides classes for handling input and output operations. It allows you to work with streams, such as reading from and writing to files, strings, and other data sources. The io module is a part of the Python Standard Library and is available in all Python installations.

docxtpl:

docxtpl is not a standard Python module but likely refers to the docxtpl library, which is a third-party library for working with Microsoft Word (.docx) files in Python. It provides a template engine that allows you to fill placeholders in a Word document template with dynamic content.

random:

The random module in Python is another built-in module that provides functions for generating random numbers and performing random selections. It is commonly used for tasks that require randomization, such as simulations, shuffling data, generating random passwords, and more.

TKINTER MODULE

TKINTER MODULE

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. Here, objects called widgets make up the window and have master or parent widget. Properties of the widgets are specified with keyword arguments.

Keyword arguments have the same name as the corresponding resource under Tk.

Widgets are positioned with one of the geometry managers Place, Pack or Grid. These managers can be called with methods place, pack, and grid available in every Widget.

The various widgets used in this program are:

Button:

The Button widget is a standard Tkinter widget, which is used for various kinds of buttons. A button is a widget which is designed for the user to interact with, i.e. if the button is pressed by mouse click some action might be started. They can also contain text and images like labels.

Entry:

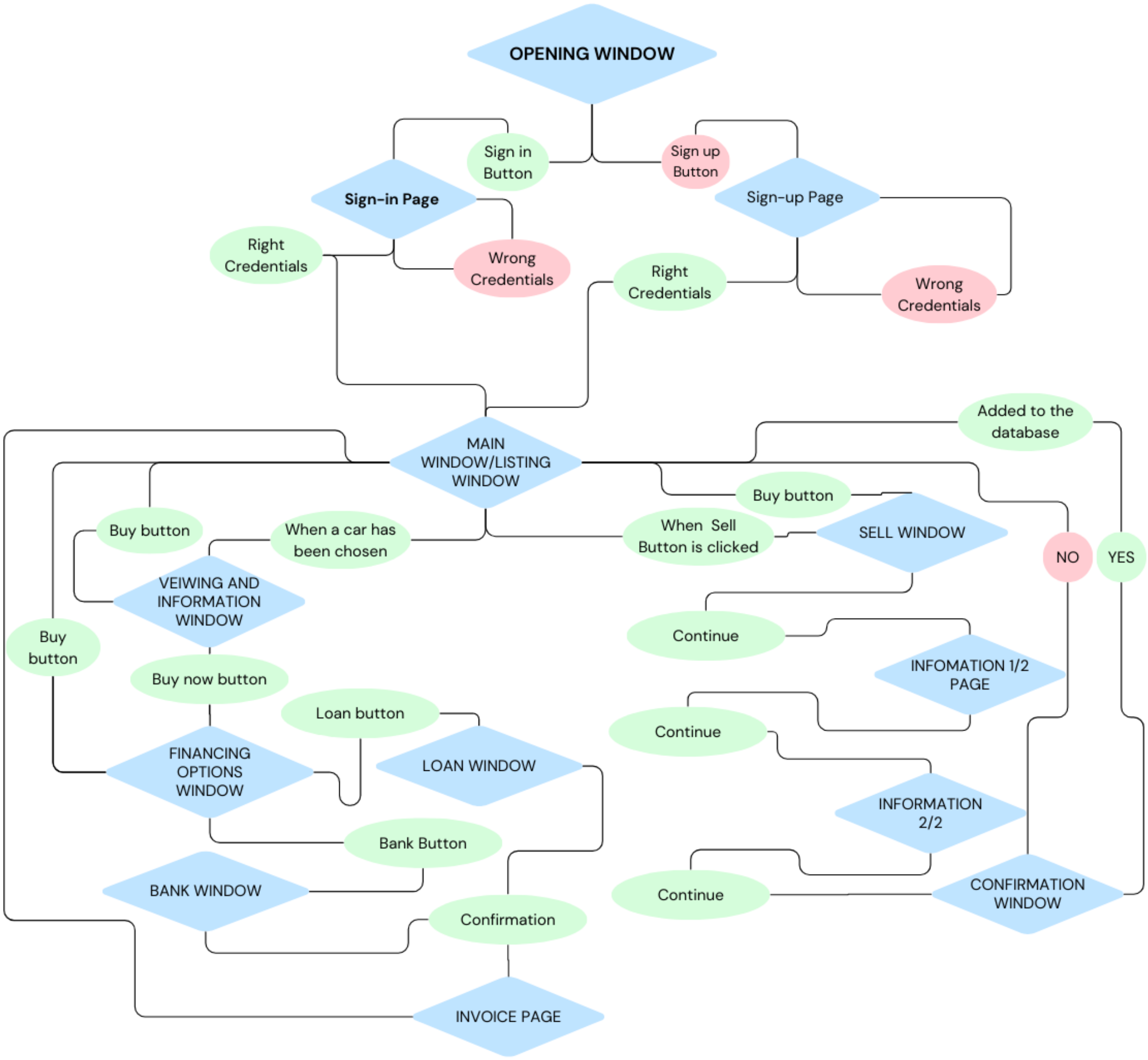
Entry widgets are the basic widgets of Tkinter used to get input, i.e. text strings, from the user of an application. This widget allows the user to enter a single line of text. If the user enters a string, which is longer than the available display space of the widget, the content will be scrolled.

Label:

This widget implements a display box where you can place text or images. The text displayed by this widget can be updated at any time you want. It is also possible to underline part of the text (like to identify a keyboard shortcut) and span the text across multiple lines.

DATA FLOW DIAGRAM

DATA FLOW DIAGRAM



MySQL

MySQL

MySQL is an open-source relational database management system (RDBMS). A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use 10 create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

ADVANTAGES:

- More secure as it consists of a solid data security layer to protect sensitive data from intruders and passwords in MySQL are encrypted.
- Available for free to download and use from the official site of MySQL.
- Is compatible with most of the operating systems, including Windows, Linux, NetWare, Novell, Solaris and other variations of UNIX.

DISADVANTAGES:

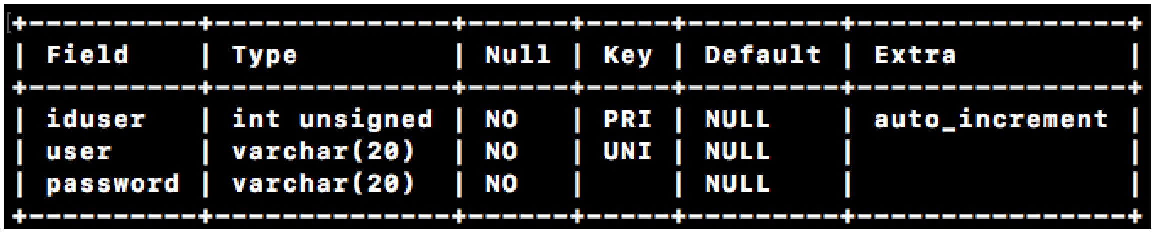
- Is not very efficient in handling very large databases.
- Is prone to data corruption as it inefficient in handling transactions
- It does not support SQL check constraints.

TABLES CREATED:

During the execution of this project, three tables were created in MySQL which can either be viewed in the text editor or through the command prompt.

USER TABLE:

TABLE: user **PRIMARY KEY**: iduser

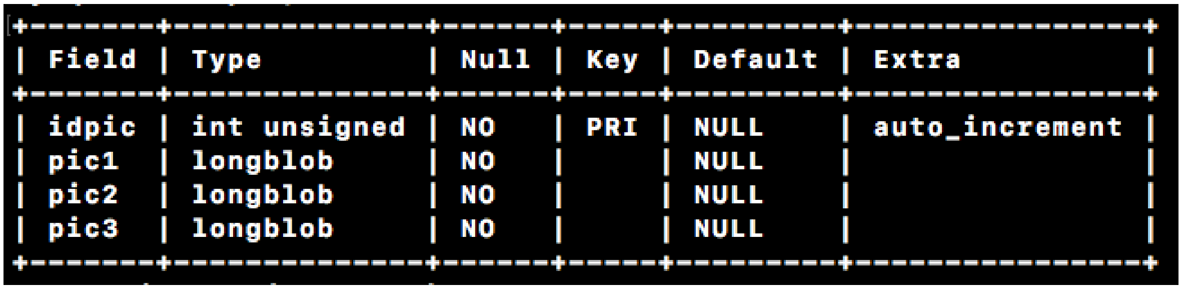


```
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| iduser   | int unsigned | NO   | PRI | NULL    | auto_increment |
| user     | varchar(20)  | NO   | UNI | NULL    |                |
| password | varchar(20)  | NO   |     | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
```

This table was created to store the login details of the user, along with their username and password. When the user logs in with their respective username and password, they are checked against this database.

PIC TABLE:

TABLE: pic **PRIMARY KEY**: idpic

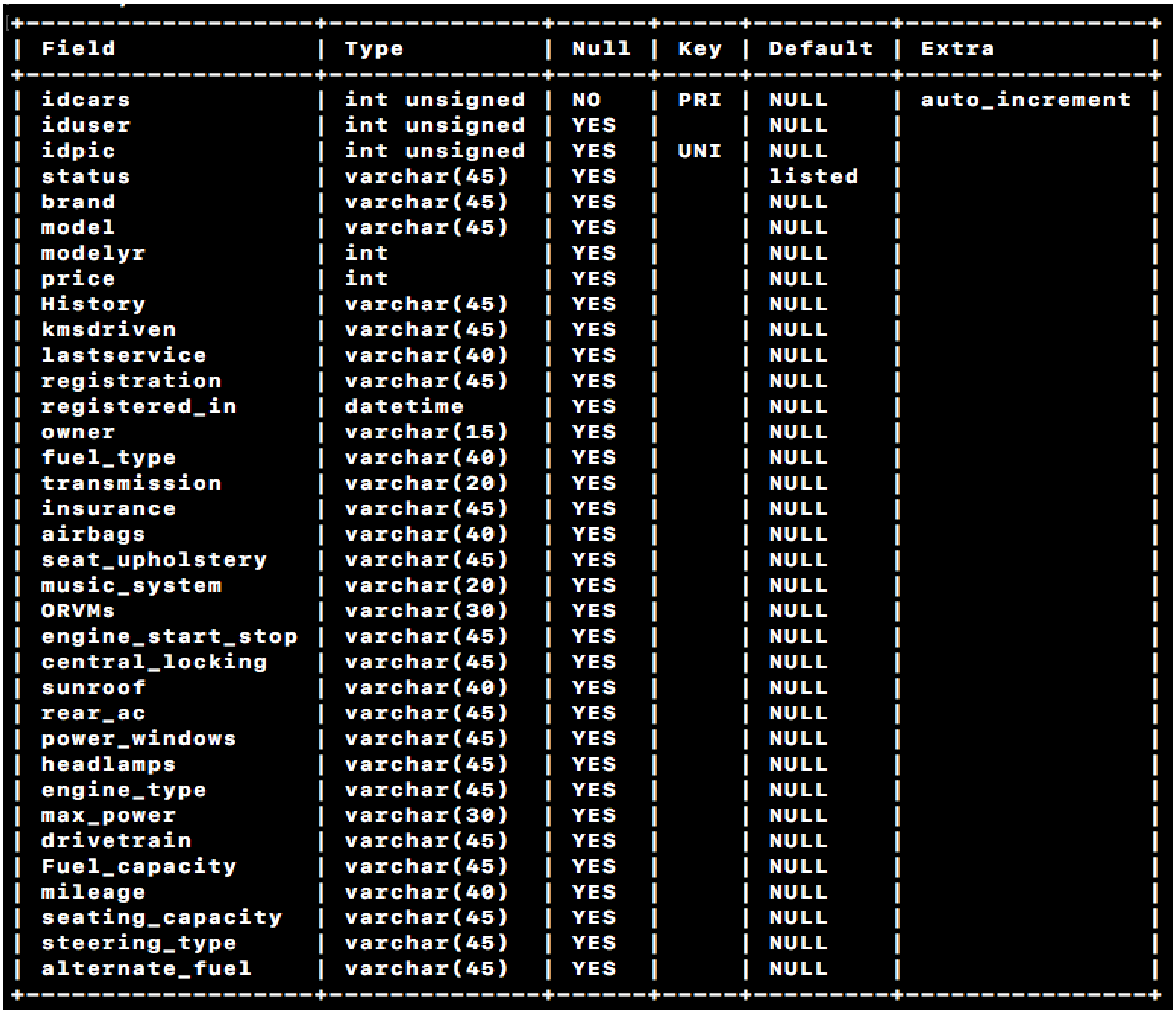


```
+-------+--------------+------+-----+---------+----------------+
| Field | Type         | Null | Key | Default | Extra          |
+-------+--------------+------+-----+---------+----------------+
| idpic | int unsigned | NO   | PRI | NULL    | auto_increment |
| pic1  | longblob     | NO   |     | NULL    |                |
| pic2  | longblob     | NO   |     | NULL    |                |
| pic3  | longblob     | NO   |     | NULL    |                |
+-------+--------------+------+-----+---------+----------------+
```

This table was created to store the blob photos of the cars.

Details_of TABLE:

TABLE: details_of **PRIMARY KEY**: idcars



```
+-------------------+--------------+------+-----+---------+----------------+
| Field             | Type         | Null | Key | Default | Extra          |
+-------------------+--------------+------+-----+---------+----------------+
| idcars            | int unsigned | NO   | PRI | NULL    | auto_increment |
| iduser            | int unsigned | YES  |     | NULL    |                |
| idpic             | int unsigned | YES  | UNI | NULL    |                |
| status            | varchar(45)  | YES  |     | listed  |                |
| brand             | varchar(45)  | YES  |     | NULL    |                |
| model             | varchar(45)  | YES  |     | NULL    |                |
| modelyr           | int          | YES  |     | NULL    |                |
| price             | int          | YES  |     | NULL    |                |
| History           | varchar(45)  | YES  |     | NULL    |                |
| kmsdriven         | varchar(45)  | YES  |     | NULL    |                |
| lastservice       | varchar(40)  | YES  |     | NULL    |                |
| registration      | varchar(45)  | YES  |     | NULL    |                |
| registered_in     | datetime     | YES  |     | NULL    |                |
| owner             | varchar(15)  | YES  |     | NULL    |                |
| fuel_type         | varchar(40)  | YES  |     | NULL    |                |
| transmission      | varchar(20)  | YES  |     | NULL    |                |
| insurance         | varchar(45)  | YES  |     | NULL    |                |
| airbags           | varchar(40)  | YES  |     | NULL    |                |
| seat_upholstery   | varchar(45)  | YES  |     | NULL    |                |
| music_system      | varchar(20)  | YES  |     | NULL    |                |
| ORVMs             | varchar(30)  | YES  |     | NULL    |                |
| engine_start_stop | varchar(45)  | YES  |     | NULL    |                |
| central_locking   | varchar(45)  | YES  |     | NULL    |                |
| sunroof           | varchar(40)  | YES  |     | NULL    |                |
| rear_ac           | varchar(45)  | YES  |     | NULL    |                |
| power_windows     | varchar(45)  | YES  |     | NULL    |                |
| headlamps         | varchar(45)  | YES  |     | NULL    |                |
| engine_type       | varchar(45)  | YES  |     | NULL    |                |
| max_power         | varchar(30)  | YES  |     | NULL    |                |
| drivetrain        | varchar(45)  | YES  |     | NULL    |                |
| Fuel_capacity     | varchar(45)  | YES  |     | NULL    |                |
| mileage           | varchar(40)  | YES  |     | NULL    |                |
| seating_capacity  | varchar(45)  | YES  |     | NULL    |                |
| steering_type     | varchar(45)  | YES  |     | NULL    |                |
| alternate_fuel    | varchar(45)  | YES  |     | NULL    |                |
+-------------------+--------------+------+-----+---------+----------------+
```

This table contains the all the details of the cars.

SOURCE CODE

SOURCE CODE

```
from pathlib import Path
from tkinter import *
from tkinter import Tk, Canvas, Entry, Text, Button, PhotoImage,filedialog,ttk,messagebox
import mysql.connector
from datetime import datetime
import tkinter as tk
import io
from mysql.connector import Error
from PIL import ImageTk, Image
import random
from docxtpl import DocxTemplate

OUTPUT_PATH = Path(__file__).parent
ASSETS_PATH = OUTPUT_PATH / Path(r"/Users/siriusjulius/Documents/ultimate 2.1/assets")

car_models = {
        "Toyota": ["Camry", "Corolla", "Rav4", "Highlander", "Tacoma", "Sienna", "Prius"],
        "Honda": ["Accord", "Civic", "CR-V", "Pilot", "Odyssey", "Fit", "HR-V"],
        "Ford": ["Mustang", "F-150", "Escape", "Explorer", "Focus", "Edge", "Ranger"],
        "Chevrolet": ["Camaro", "Silverado", "Equinox", "Tahoe", "Malibu", "Traverse", "Colorado"],
        "Volkswagen": ["Golf", "Jetta", "Passat", "Tiguan", "Atlas", "Arteon", "ID.4"],
        "BMW": ["3 Series", "5 Series", "X3", "X5", "7 Series", "X1", "4 Series"],
        "Mercedes-Benz": ["C-Class", "E-Class", "GLC", "GLE", "A-Class", "S-Class", "G-Class"],
        "Audi": ["A4", "A6", "Q5", "Q7", "Q3", "A3", "TT"],
        "Nissan": ["Altima", "Sentra", "Rogue", "Pathfinder", "Maxima", "Murano", "Titan"],
        "Hyundai": ["Elantra", "Sonata", "Tucson", "Santa Fe", "Kona", "Veloster", "Palisade"],
        "Kia": ["Optima", "Sorento", "Sportage", "Telluride", "Forte", "Soul", "Stinger"],
        "Volvo": ["S60", "XC60", "XC90", "V90", "XC40", "S90", "V60"],
        "Mazda": ["Mazda3", "Mazda6", "CX-5", "CX-9", "MX-5", "CX-3", "CX-30"],
        "Subaru": ["Impreza", "Legacy", "Outback", "Forester", "Crosstrek", "BRZ", "Ascent"],
        "Lexus": ["IS", "ES", "RX", "NX", "GS", "LS", "UX"],
        "Jeep": ["Wrangler", "Cherokee", "Grand Cherokee", "Renegade", "Compass", "Gladiator", "Wrangler
Unlimited"],
        "Tesla": ["Model S", "Model 3", "Model X", "Model Y", "Cybertruck", "Roadster", "Semi"],
        "Ferrari": ["488 GTB", "812 Superfast", "Portofino", "SF90 Stradale", "GTC4Lusso", "F8 Tributo",
"LaFerrari"],
        "Porsche": ["911", "Cayenne", "Macan", "Panamera", "Boxster", "Taycan", "Cayman"],
        "Land Rover": ["Range Rover", "Discovery", "Defender", "Velar", "Range Rover Sport", "Range Rover
Evoque", "Discovery Sport"],
        "Maruti Suzuki":["Alto","Swift","Dzire","Baleno","Wagon R","Ertiga","Vitara Brezza"
]
}
car_brands = {'brand':["Toyota","Maruti Suzuki","Honda","Ford","Chevrolet","Volkswagen","BMW","Mercedes-
Benz","Audi","Nissan","Hyundai","Kia","Volvo","Mazda","Subaru","Lexus","Jeep","Tesla","Ferrari","Porsche","La
nd Rover"],
'model_yr':list(range( int(datetime.now().year),1999,-1)),
'rto':["AP - Andhra Pradesh","AR - Arunachal Pradesh","AS - Assam","BR - Bihar","CG - Chhattisgarh","DL -
Delhi","GA - Goa","GJ - Gujarat","HR - Haryana","HP - Himachal Pradesh","JK - Jammu and Kashmir","KA -
Karnataka","KL - Kerala","MP - Madhya Pradesh","MH - Maharashtra","MN - Manipur","ML - Meghalaya","MZ -
Mizoram","NL - Nagaland","OD - Odisha","PB - Punjab","RJ - Rajasthan","SK - Sikkim","TN - Tamil Nadu","TS -
Telangana","TR - Tripura","UP - Uttar Pradesh","UK - Uttarakhand","WB - West Bengal","AN - Andaman and
Nicobar Islands","CH - Chandigarh","DN - Dadra and Nagar Haveli","DD - Daman and Diu","LD -
Lakshadweep","PY - Puducherry"],
'kmsdriven': ['Less than 10,000 km', '10,000 - 25,000 km', '25,000 - 50,000 km', '50,000 - 100,000 km', 'More than
100,000 km']}

car_attributes = {
        'History': ['Clean', 'Accident-free', 'Service records available', 'One owner', 'Non-smoker'],
        'kmsdriven': ['Less than 10,000 km', '10,000 - 25,000 km', '25,000 - 50,000 km', '50,000 - 100,000 km',
'More than 100,000 km'],
        'lastservice': ['Less than 3 months ago', '3 - 6 months ago', '6 - 12 months ago', 'Over a year ago', 'No service
history'],
        'registration': ['input a date'],
        'registered_in': ["AP - Andhra Pradesh","AR - Arunachal Pradesh","AS - Assam","BR - Bihar","CG -
Chhattisgarh","DL - Delhi","GA - Goa","GJ - Gujarat","HR - Haryana","HP - Himachal Pradesh","JK - Jammu and
Kashmir","KA - Karnataka","KL - Kerala","MP - Madhya Pradesh","MH - Maharashtra","MN - Manipur","ML -
Meghalaya","MZ - Mizoram","NL - Nagaland","OD - Odisha","PB - Punjab","RJ - Rajasthan","SK - Sikkim","TN -
Tamil Nadu","TS - Telangana","TR - Tripura","UP - Uttar Pradesh","UK - Uttarakhand","WB - West Bengal","AN -
Andaman and Nicobar Islands","CH - Chandigarh","DN - Dadra and Nagar Haveli","DD - Daman and Diu","LD -
Lakshadweep","PY - Puducherry"],
        'owner': ['First', 'Second', 'Third or more', 'Company owned'],
        'fuel_type': ['Petrol', 'Diesel', 'CNG', 'LPG', 'Electric'],
        'transmission': ['Manual', 'Automatic', 'CVT', 'AMT', 'Dual-clutch'],
        'insurance': ['Valid', 'Expired', 'Renewed recently', 'Pending renewal', 'No insurance'],
        'airbags': ['Driver', 'Driver + Passenger', 'Driver + Passenger + Side', 'Driver + Passenger + Side + Curtain',
'No airbags'],
        'seat_upholstery': ['Leather', 'Fabric', 'Artificial leather', 'Velvet', 'Suede'],
        'music_system': ['Basic', 'CD player + FM/AM', 'Touchscreen + Bluetooth + USB', 'Navigation + Apple
CarPlay/Android Auto', 'Premium brand'],
        'ORVMs': ['Manual adjustment', 'Power adjustment', 'Power folding', 'Auto-dimming', 'Heated'],
        'engine_start_stop': ['Not available', 'Manual', 'Push button', 'Smart key', 'Remote'],
        'central_locking': ['Manual', 'Remote', 'Smart key', 'Automatic', 'Child lock'],
        'sunroof': ['Not available', 'Manual', 'Electric', 'Panoramic', 'Tilt and slide'],
        'rear_ac': ['Not available', 'Manual', 'Automatic', 'Independent controls', 'Blower'],
        'power_windows': ['Front', 'Rear', 'All four', 'One-touch up/down', 'Anti-pinch'],
        'headlamps': ['Halogen', 'Projector', 'LED', 'Xenon', 'Matrix LED'],
        'engine_type': ['2-cylinder', '3-cylinder', '4-cylinder', '5-cylinder', '6-cylinder'],
        'max_power': ['Less than 75 bhp', '75 - 100 bhp', '100 - 125 bhp', '125 - 150 bhp', 'More than 150 bhp'],
        'drivetrain': ['Front-wheel drive', 'Rear-wheel drive', 'All-wheel drive', 'Four-wheel drive', 'Electric motor'],
        'Fuel_capacity': ['Less than 30 liters', '30 - 40 liters', '40 - 50 liters', '50 - 60 liters', 'More than 60 liters'],
        'mileage': ['Less than 10 kmpl', '10 - 15 kmpl', '15 - 20 kmpl', '20 - 25 kmpl', 'More than 25 kmpl'],
        'seating_capacity': ['2-seater', '4-seater', '5-seater', '7-seater', '8-seater or more'],
        'steering_type': ['Manual', 'Power-assisted', 'Electric-assisted', 'Tilt-adjustable', 'Telescopic-adjustable'],
        'alternate_fuel': ['Hybrid', 'Plug-in Hybrid', 'Hydrogen', 'Ethanol', 'Biodiesel']}

car_brands_prices = {
        "Toyota": 1000000,
        "Honda": 950000,
        "Ford": 900000,
        "Chevrolet": 850000,
        "Volkswagen": 800000,
        "BMW": 2500000,
        "Mercedes-Benz": 3000000,
        "Audi": 2800000,
        "Nissan": 850000,
        "Hyundai": 750000,
		"Kia": 800000,
		"Volvo": 4000000,
		"Mazda": 800000,
		"Subaru": 900000,
		"Lexus": 3500000,
		"Jeep": 1200000,
		"Tesla": 5000000,
		"Ferrari": 25000000,
		"Porsche": 6000000,
		"Land Rover": 4500000,
		"Maruti Suzuki": 600000
}
buywin_view1num=0
buywin_view2num=0
buywin_view3num=0
buywin_view4num=0
buywin_view5num=0
buywin_view6num=0
inputlist= []

tableprice ="
tablebrand ="
tablemodel ="
tablemodelyr ="
tableHistory ="
tablekmsdriven ="
tablelastservice ="
tableregistration ="
tableregistered_in ="
tableowner ="
tablefuel_type ="
tabletransmission ="
tableinsurance ="
tableairbags ="
tableseat_upholstery ="
tablemusic_system ="
tableORVMs ="
tableengine_start_stop ="
tablecentral_locking ="
tablesunroof ="
tablerear_ac ="
tablepower_windows ="
tableheadlamps ="
tableengine_type ="
tablemax_power ="
tabledrivetrain ="
tableFuel_capacity ="
tablemileage ="
tableseating_capacity ="
tablesteering_type ="
tablealternate_fuel ="
varcuimgpath=[]
def on_brandcombo_select(event):
		selected_item = brandcombo.get()
		modelcombo['values'] = []
		modelcombo['values'] = car_models.get(selected_item)
```

```
def on_bbrandcombo_select(event):
        selected_item = bbrandcombo.get()
        bmodelcombo['values'] = []
        bmodelcombo['values'] = car_models.get(selected_item)

tablename=""
tablemobile=""
def open_popup():
        popup = Toplevel()
        popup.geometry("507x664")
        popup.resizable(False,False)
        popup.configure(bg="#FFFFFF")

        confirmation = Canvas(
        popup,
        bg="#FFFFFF",
        height=664,
        width=507,
        bd=0,
        highlightthickness=0,
        relief="ridge"
        )
        confirmation.place(x=0, y=0)

        image_image_1 = PhotoImage(file=relative_to_assets("popup.png"))
        image_1 = confirmation.create_image(253.0, 332.0, image=image_image_1)

        namecon = Entry(
        popup,
        bd=0,
        bg="#FFF7D8",
        fg="#000716",
        highlightthickness=0
        )
        namecon.place(x=131.0, y=213.0, width=255.0, height=31.0)

        mobilecon = Entry(
        popup,
        bd=0,
        bg="#FFF7D8",
        fg="#000716",
        highlightthickness=0
        )
        mobilecon.place(x=131.0, y=303.0, width=255.0, height=31.0)

        residencecon = Entry(
        popup,
        bd=0,
        bg="#FFF7D8",
        fg="#000716",
        highlightthickness=0
        )
        residencecon.place(x=131.0, y=385.0, width=255.0, height=31.0)

        passwordcon = Entry(
        popup,
        bd=0,
        bg="#FFF7D8",
        fg="#000716",
        highlightthickness=0
        )
        passwordcon.place(x=131.0, y=466.0, width=255.0, height=31.0)
        def confirm_input():
        name = namecon.get()
        mobile = mobilecon.get()
        residence = residencecon.get()
        password_text = passwordcon.get()
        global tablename,tablemobile
          print("Name:", name)
        print("Mobile:", mobile)
        print("Residence:", residence)
        print("Password:", password_text)
        if password_text == password:
        if mobile.isnumeric():
        tablename = name
          tablemobile = mobile
        invoiceprep('Bank')
        invoice.place(x=0,y=0)
        fin_b.place(x=3000,y=3000)
        try:
                namecon.delete(0, tk.END)
        except:
                mobilecon.delete(0, tk.END)
                residencecon.delete(0, tk.END)
                password.delete(0, tk.END)
        popup.destroy()

        else:
        messagebox.showerror("Mobile Number Error",'mobile number only should have numbers')

        else:
        messagebox.showerror("Password Error",'wrong password try again')
        confirmconbutton_image_1 = PhotoImage(file=relative_to_assets("confirmcon.png"))
        confirmcon = Button(
        popup,
         text="Confirm",
        command=confirm_input,
        image=confirmconbutton_image_1
        )
        confirmcon.place(x=152.0, y=547.0, width=204.89328002929688, height=53.0)
        cancelconbutton_image_2 = PhotoImage(file=relative_to_assets("cancelcon.png"))
        cancelcon = Button(
        popup,
        text="Cancel",
        command=popup.destroy,
        image= cancelconbutton_image_2
        )
        cancelcon.place(x=190.0, y=616.0, width=129.0, height=26.0)

        popup.mainloop()

def cur_invoice(cur):
```

```
            if cur == fin_l:
            method = "Loan"
            invoice.place(x=0,y=0)
            cur.place(x=3000,y=3000)
            else:
            method = "Bank"
            open_popup()


def invoiceprep(method):
            invoice.itemconfigure(iname,text=tablename )
            invoice.itemconfigure(imobileno,text= tablemobile)
            invoice.itemconfigure(iregistration,text=tableregistration )
            invoice.itemconfigure(imake,text=tablebrand )
            invoice.itemconfigure(imodelyr,text=tablemodelyr )
            invoice.itemconfigure(imodel,text=tablemodel )
            invoice.itemconfigure(imileage,text=tablemileage )
            invoice.itemconfigure(imethod,text=method )
            invoice.itemconfigure(iprice,text=('₹',tableprice) )
            invoice.itemconfigure(itotal,text=('₹',tableprice + tableprice*0.1) )
            invoice.itemconfigure(itax,text=('₹',tableprice/10,'(10%)') )
            print(tableprice + tableprice*0.2)
            print(tableprice*0.1,'(10%)')
def adduser_data(u,p):
    conn = mysql.connector.connect(
            host='localhost',
            user='root',
            password='data@6420',
            database='cars'
)
            cursor = conn.cursor()
            cursor.execute(f"INSERT INTO user(user,password) VALUES({u},{p})")
            conn.commit()
#           conn.close()
user = ""
password =""
def check_user_credentials(user, password):
            conn = mysql.connector.connect(
            host="localhost",
            user="root",
            password="data@6420",
            database="cars"
            )
            c = conn.cursor()
            query = "SELECT * FROM user WHERE user = %s"
            values = (user,)
            c.execute(query, values)

            row = c.fetchone()
            if row is not None:
            print(row)
            if row[2] == password:
            messagebox.showinfo("Success", "Welcome, " + user + "!")
            cur_buy(sgninwin)
            else:
            messagebox.showerror("Error", "Incorrect password for user " + user + ".")
```

```
        else:
        messagebox.showerror("Error", "User " + user + " not found.")

        conn.close()

def sign_func(u,p,r):
        print(u,p,r)
        if len(u)<5 or len(p)<5:
        messagebox.showerror("Yikes",'username or the password should have minimum length of 5 characters')
        elif p!=r:
        messagebox.showerror("Yikes", 'the password does not match!')
        else:
        adduser_data(u,p)
        messagebox.showinfo('successfull','a new account has been created').f
        cur_buy(sgnupwin)

def relative_to_assets(path: str) -> Path:
        return ASSETS_PATH / Path(path)


def cur_slwin(current):
        if current != sel_1 and current != sel_2:
        state = True
        else:
        state = False
        if state == True:
        current.place(x=3000,y=3000)
        slwin.place(x=0,y=0)
        clearvarsel()
        else:
        result = messagebox.askquestion("Confirmation", "Do you want to proceed?")
        if result == 'yes':
        print("User clicked 'Yes'")

        current.place(x=3000,y=3000)
        slwin.place(x=0,y=0)
        clearvarsel()
        else:
        pass

def clearvarsel():
        global tableprice ,tablebrand ,tablemodel ,tablemodelyr ,tableHistory ,tablekmsdriven ,tablelastservice
,tableregistration ,tableregistered_in ,tableowner ,tablefuel_type ,tabletransmission ,tableinsurance ,tableairbags
,tableseat_upholstery ,tablemusic_system ,tableORVMs ,tableengine_start_stop ,tablecentral_locking ,tablesunroof
,tablerear_ac ,tablepower_windows ,tableheadlamps ,tableengine_type ,tablemax_power ,tabledrivetrain
,tableFuel_capacity ,tablemileage ,tableseating_capacity ,tablesteering_type ,tablealternate_fuel
        var1= [tableprice ,tablebrand ,tablemodel ,tablemodelyr ,tableHistory ,tablekmsdriven ,tablelastservice
,tableregistration ,tableregistered_in ,tableowner ,tablefuel_type ,tabletransmission ,tableinsurance ,tableairbags
,tableseat_upholstery ,tablemusic_system ,tableORVMs ,tableengine_start_stop ,tablecentral_locking ,tablesunroof
,tablerear_ac ,tablepower_windows ,tableheadlamps ,tableengine_type ,tablemax_power ,tabledrivetrain
,tableFuel_capacity ,tablemileage ,tableseating_capacity ,tablesteering_type ,tablealternate_fuel]
        varsel =[brandcombo,kmsdrivencombo,model_yrcombo,modelcombo,rtocombo]
        varsl_1=[sl_1airbagscombo,sl_1drivetraincombo,sl_1enginetypecombo,sl_1fueltypecombo,sl_1historycom
bo,sl_1insurancecombo,sl_1kmdrivencombo,sl_1last_servicecombo,sl_1mileagecombo,sl_1ownercombo,sl_1steerin
gcombo,sl_1transmissiontypecombo,sl_1seatupholstrycombo,sl_1registeredincombo,sl_1registrationcombo]
```

```
	varsl_2=[orvmscombo,enginestart_topcombo,centrallockingcombo,sunroofcombo,rearaccombo,powerwind
owscombo,headlampscombo,maxpowercombo,fueltankcapcombo,seatingcapacitycombo,alterantefueltypecombo,mu
sicsystemcombo,transmissioncombo]
	print(len(varsl_1))
	print(len(varsl_2))
	print(len(var1))
	tableprice ,tablebrand ,tablemodel ,tablemodelyr ,tableHistory ,tablekmsdriven ,tablelastservice
,tableregistration ,tableregistered_in ,tableowner ,tablefuel_type ,tabletransmission ,tableinsurance ,tableairbags
,tableseat_upholstery ,tablemusic_system ,tableORVMs ,tableengine_start_stop ,tablecentral_locking ,tablesunroof
,tablerear_ac ,tablepower_windows ,tableheadlamps ,tableengine_type ,tablemax_power ,tabledrivetrain
,tableFuel_capacity ,tablemileage ,tableseating_capacity ,tablesteering_type ,tablealternate_fuel =
",",",",",",",",",",",",",",",",",",",",",",",",",",",","

	for i in varsel :
	i.set(")
	for i in varsl_1 :
	if i == sl_1registeredincombo:
	i.delete(0, tk.END)
	else:
	i.set(")
	for i in varsl_2 :
	i.set(")
def cur_sgnin(current):
	current.place(x=3000,y=3000)
	sgninwin.place(x=0,y=0)

def cur_sgnup(current):
	current.place(x=3000,y=3000)
	sgnupwin.place(x=0,y=0)
random_items=[]
def getidcars():
	mydb = mysql.connector.connect(
	host="localhost",
	user="root",
	password="data@6420",
	database="cars"
)
	cursor = mydb.cursor()

	query = "SELECT idcars FROM details_of where status = 'listed'"
	cursor.execute(query)

	result = cursor.fetchall()

	for row in result:
	idcar = row[0]
	idcar_list = [row[0] for row in result]
	print(idcar_list)
	import random
	global random_items
	random_items = random.sample(idcar_list, len(idcar_list))

def assigncarsbutton(slide):
        global
random_items,buywin_view1num,buywin_view2num,buywin_view3num,buywin_view4num,buywin_view5num,bu
ywin_view6num
        if slide == 1:
        index =0
        l = []
        for i in
(buywin_view1num,buywin_view2num,buywin_view3num,buywin_view4num,buywin_view5num,buywin_view6n
um):
        print (index)
        try:
        l.append(random_items[index])
        index+=1
        except IndexError:
        l.append(random_items[0])
        index+=1

        buywin_view1num,buywin_view2num,buywin_view3num,buywin_view4num,buywin_view5num,buywin_
view6num = l

        print([buywin_view1num,buywin_view2num,buywin_view3num,buywin_view4num,buywin_view5num,bu
ywin_view6num])
        asssignpicprog()
        elif slide == 2:
        l = []
        index =6
        for i in
[buywin_view1num,buywin_view2num,buywin_view3num,buywin_view4num,buywin_view5num,buywin_view6n
um]:
        try:
        l.append(random_items[index])
        index+=1
        except IndexError:
          l.append(random_items[0])

        buywin_view1num,buywin_view2num,buywin_view3num,buywin_view4num,buywin_view5num,buywin_
view6num = l
        asssignpicprog()
        elif slide == 3:
        l = []
        index =12
        for i in
[buywin_view1num,buywin_view2num,buywin_view3num,buywin_view4num,buywin_view5num,buywin_view6n
um]:
        try:
        l.append(random_items[index])
        index+=1
        except IndexError:
        l.append(random_items[0])

        buywin_view1num,buywin_view2num,buywin_view3num,buywin_view4num,buywin_view5num,buywin_
view6num = l
        asssignpicprog()
```

```
def asssignpicprog():
    num = 0
    x = [buywin_view1,buywin_view2,buywin_view3,buywin_view4,buywin_view5,buywin_view6]
    lis =
[buywin_view1num,buywin_view2num,buywin_view3num,buywin_view4num,buywin_view5num,buywin_view6n
um]
    print(lis)
    buttonlist= [abutton,cbutton,bbutton,dbutton,fbutton,ebutton]
    for i in
[buywin_view1num,buywin_view2num,buywin_view3num,buywin_view4num,buywin_view5num,buywin_view6n
um]:
    cursor = mydb.cursor()
    print('i',i)
    query=f"SELECT brand, model FROM details_of WHERE idcars = {i}"
    cursor.execute(query)
    result = cursor.fetchall()
    result = result[0]
    buttonlist[num].configure(text=(f"{result[0]} {result[1]}"),width=200,font=("TkDefaultFont", 18))
    query = f"SELECT idpic FROM details_of where idcars = {i}"
    cursor.execute(query)
    result = cursor.fetchall()
    print('result',result)
    result = result[0][0]
    sql = "SELECT pic1 FROM pic WHERE idpic = %s"
    values = (result,)
    abuton = Button(buywin)
    abuton.place
    mycursor.execute(sql, values)
    result = mycursor.fetchone()
    imgtk = ImageTk.PhotoImage(Image.open(io.BytesIO(result[0])).resize((457, 220)))
    x[num].configure(image=imgtk)
    x[num].image = imgtk
    num+=1




bankchosen =""
def cur_buy(current):
    global
buywin_view1num,buywin_view2num,buywin_view3num,buywin_view4num,buywin_view5num,buywin_view6nu
m
    varbuy=[brandcombo,kmsdrivencombo,model_yrcombo,modelcombo,rtocombo]
    a =
[buywin_view1num,buywin_view2num,buywin_view3num,buywin_view4num,buywin_view5num,buywin_view6n
um]
    for j in a:
      j = 0
    for i in varbuy :
    i.set('')
    if current != sel_1 and current != sel_2:
    state = True
    else:
    state = False
    if state == True:
    current.place(x=3000,y=3000)
    buywin.place(x=0,y=0)
    getidcars()
    assigncarsbutton(1)
    clearvarsel()
    else:
    result = messagebox.askquestion("Confirmation", "Do you want to proceed?")
    if result == 'yes':
    print("User clicked 'Yes'")
    current.place(x=3000,y=3000)
   buywin.place(x=0,y=0)
    getidcars()
    assigncarsbutton(1)
    clearvarsel()
    else:
    pass
def cur_buynow(current):
    global tablebrand,tablemodel,tablemodelyr,tableprice
    current.place(x=3000,y=3000)
    fin_s.place(x=0,y=0)
    fin_s.itemconfigure(car_namef, text=('Car:',tablebrand,tablemodel,tablemodelyr) )
    fin_s.itemconfigure(car_namef2,text=(tablebrand,tablemodel,tablemodelyr))
    fin_s.itemconfigure(car_pricef,text= ('₹',tableprice))
    fin_b.itemconfigure(price_nd_nameb,text= (f"₹{tableprice} + Tax ({tablebrand} {tablemodel}
{tablemodelyr})"))
    fin_l.itemconfigure(price_nd_namel,text= (f"₹{tableprice} + Tax ({tablebrand} {tablemodel}
{tablemodelyr})"))

def fin_s_respect(respect):
    fin_s.place(x=3000,y=3000)
    respect.place(x=0,y=0)
def reveal_password(password_entry,eye_):
    if password_entry.cget("show") == "":
    password_entry.config(show="*")
    if eye_ == eye_sgnupwin:
    eye_sgnupwin_img.configure(file=relative_to_assets("eye_sgnupwin.png"))
    elif eye_== eye_sgninwin:
    eye_sgninwin_img.configure(file=relative_to_assets("eye_sgninwin.png"))

    else:
    password_entry.config(show="")
    if eye_ == eye_sgnupwin:
    eye_sgnupwin_img.configure(file=relative_to_assets("eye_sgnupwin1.png"))
    elif eye_== eye_sgninwin:
    eye_sgninwin_img.configure(file=relative_to_assets("eye_sgninwin1.png"))
picid_chosen= 0
def cur_buywin1(button):
    global picid_chosen,tableprice ,tablebrand ,tablemodel ,tablemodelyr ,tableHistory ,tablekmsdriven
,tablelastservice ,tableregistration ,tableregistered_in ,tableowner ,tablefuel_type ,tabletransmission ,tableinsurance
,tableairbags ,tableseat_upholstery ,tablemusic_system ,tableORVMs ,tableengine_start_stop ,tablecentral_locking
,tablesunroof ,tablerear_ac ,tablepower_windows ,tableheadlamps ,tableengine_type ,tablemax_power
,tabledrivetrain ,tableFuel_capacity ,tablemileage ,tableseating_capacity ,tablesteering_type ,tablealternate_fuel
    buywin.place(x=3000,y=3000)
    buywin_1.place(x=0,y=0)
    switch_buywin_specs(buywin_1_overview,overviewcanvas)
    carid = button
    picid_chosen = carid
```

```
        cursor = mydb.cursor()
        column_names = ['price','brand','model','modelyr','History', 'kmsdriven', 'lastservice', 'registration',
'registered_in', 'owner',
                'fuel_type', 'transmission', 'insurance', 'airbags', 'seat_upholstery', 'music_system', 'ORVMs',
                'engine_start_stop', 'central_locking', 'sunroof', 'rear_ac', 'power_windows', 'headlamps',
'engine_type',
                'max_power', 'drivetrain', 'Fuel_capacity', 'mileage', 'seating_capacity', 'steering_type',
'alternate_fuel']



        query = f"SELECT {', '.join(column_names)} from details_of where idcars = {carid} "
        cursor.execute(query)
        rows = cursor.fetchone()
        tableprice ,tablebrand ,tablemodel ,tablemodelyr ,tableHistory ,tablekmsdriven ,tablelastservice
,tableregistration ,tableregistered_in ,tableowner ,tablefuel_type ,tabletransmission ,tableinsurance ,tableairbags
,tableseat_upholstery ,tablemusic_system ,tableORVMs ,tableengine_start_stop ,tablecentral_locking ,tablesunroof
,tablerear_ac ,tablepower_windows ,tableheadlamps ,tableengine_type ,tablemax_power ,tabledrivetrain
,tableFuel_capacity ,tablemileage ,tableseating_capacity ,tablesteering_type ,tablealternate_fuel = rows
#       cursor.close()
#       mydb.close()
        var1= [tableprice ,tablebrand ,tablemodel ,tablemodelyr ,tableHistory ,tablekmsdriven ,tablelastservice
,tableregistration ,tableregistered_in ,tableowner ,tablefuel_type ,tabletransmission ,tableinsurance ,tableairbags
,tableseat_upholstery ,tablemusic_system ,tableORVMs ,tableengine_start_stop ,tablecentral_locking ,tablesunroof
,tablerear_ac ,tablepower_windows ,tableheadlamps ,tableengine_type ,tablemax_power ,tabledrivetrain
,tableFuel_capacity ,tablemileage ,tableseating_capacity ,tablesteering_type ,tablealternate_fuel]
        print(var1)

        name = f"{tablebrand} {tablemodel} {tablemodelyr}"
        buywin_1.itemconfigure(buy_c_name,text = name)
        buywin_1.itemconfigure(buy_c_kms,text = (tablekmsdriven))
        buywin_1.itemconfigure(buy_c_fuel_t,text = (tablefuel_type))
        buywin_1.itemconfigure(buy_c_regis,text = (tableregistration[:2]))
        buywin_1.itemconfigure(buy_c_price,text = ('₹',tableprice))
        fin_l.itemconfigure(bankfirml,text= "Choose a bank")
        fin_b.itemconfigure(bankfirmb,text= "Choose a bank")
        try:
        buywin_1.itemconfigure(buy_c_downpayment,text = ('₹',tableprice//60))
        except:
        pass
        overlis =
[bhistory,bkms,blst_service,bregistration,bregisteredin,bowner,bfuel_type,btransmission,binsurance]
        overlis2 = [tableHistory,tablekmsdriven,tablelastservice,tableregistration,tableregistered_in.strftime("%d-
%m-%Y"),tableowner,tablefuel_type,tabletransmission,tableinsurance]
        num = 0
        for i in overlis :
        overviewcanvas.itemconfigure(i,text=overlis2[num])
        num+=1
        flis =
[fairbags,fseatupholstry,fmusicsys,fovrm,fenginestrt,fcentralock,fsunroof,frearac,fpowrwind,fheadlamps]
        flis2 =
[tableairbags,tableseat_upholstery,tablemusic_system,tableORVMs,tableengine_start_stop,tablecentral_locking,table
sunroof,tablerear_ac,tablepower_windows,tableheadlamps]
        num = 0
        for i in flis :
        featurecanvas.itemconfigure(i,text=flis2[num])
```

```
    num+=1
    speclis = [specfuel_type,specengine_type ,specdrivetrain ,specmileage ,specsteeringtyp
,spectransmissiontyp ,specmaxpwr ,specfueltank ,specseatincap ,speclternatefueltype ]
    speclis2 =
[tablefuel_type,tableengine_type,tabledrivetrain,tablemileage,tablesteering_type,tabletransmission,tablemax_power,t
ableFuel_capacity,tableseating_capacity,tablealternate_fuel]
    num = 0
    for i in speclis :
    specificationscanvas.itemconfigure(i,text=speclis2[num])
    num+=1

    x = [buywin_1_veiw2_4,buywin_1_veiw3_5,buywin_1_veiw4_6]
    query = f"SELECT idpic FROM details_of where idcars = {carid}"
    cursor.execute(query)
    result = cursor.fetchall()
    print('result',result)
    result = result[0][0]
    picid_chosen = result
    sql = "SELECT pic1,pic2,pic3 FROM pic WHERE idpic = %s"
    values = (result,)
    mycursor.execute(sql, values)
    result = mycursor.fetchmany()
    #size

    imgtk = ImageTk.PhotoImage(Image.open(io.BytesIO(result[0][0])).resize((830, 400)))
    buywin_1_veiw1_3.configure(image=imgtk)
    buywin_1_veiw1_3.image = imgtk
    imgtk = ImageTk.PhotoImage(Image.open(io.BytesIO(result[0][0])).resize((950, 520)))#hdbhdsbf
    fin_sbutton_5.configure(image=imgtk)
    fin_sbutton_5.image = imgtk
    num = 0
    for i in result[0]:
    image = Image.open(io.BytesIO(i))
    resized_image = image.resize((226, 117), resample=Image.LANCZOS)
    imgtk = ImageTk.PhotoImage(resized_image)
        x[num].configure(image=imgtk)
    x[num].image = imgtk
    #imgtk = ImageTk.PhotoImage(Image.open(io.BytesIO(i)).resize((226, 117)))
    #x[num].configure(image=imgtk)
    #x[num].image = imgtk
    num += 1

def switch_sel(swithbtn):

    if swithbtn==switch2_buywin:
    assigncarsbutton(2)
    switch1_buywin.place(x=881,y=1040,width=16,height=16
    )
    switch3_buywin.place(x=1009,y=1040,width=16,height=16
    )
    switch2_buywin.place(x=3000,y=1040,width=16,height=16
    )


    elif swithbtn==switch3_buywin:
    switch1_buywin.place(x=881,y=1040,width=16,height=16
    )
    switch3_buywin.place(x=3000,y=1040,width=16,height=16
    )
    switch2_buywin.place(x=945,y=1040,width=16,height=16
    )
    assigncarsbutton(3)
    elif swithbtn==switch1_buywin:
    switch1_buywin.place(x=3000,y=1040,width=16,height=16
    )
    switch3_buywin.place(      x=1009, y=1040, width=16,           height=16)
    switch2_buywin.place(x=945,y=1040,width=16,height=16)
    assigncarsbutton(1)

def buywin_1_buywin(current):
    current.place(x=3000,y=3000)
    buywin.place(x=0,y=0)
def switch_buywin(switch):
    a = [switch1_buywin,switch2_buywin,switch3_buywin]
 a = [buywin_1_specification]
    x=0
    b=[[881,1040],[945,1040],[1009,1040]]
    for i in a :
    if i == switch:
    i.place(x=3000,y=3000)
    x+=1
    else:
    i.place(x=b[x][0],y=b[x][1])
    x+=1

def invoicedownload():
    global tablename,tablemobile,tablemodelyr,tablemodel,tableprice,tablebrand
    doc = DocxTemplate('/Users/siriusjulius/Documents/ultimate 2.1/carinvoiceoutline.docx')
    doc.render({'cust_name':tablename,'mobile_num':tablemobile,"date_purchase":'18 June
2023','residence_cust':'ABCD
india',"carmodel":(tablebrand,tablemodel),"mdlyr":tablemodelyr,'$'+str(tableprice):'$700000',"taxprice":'$'+str(tablep
rice/10),'ttlprice':'$'+str(tableprice+tableprice*0.1),'price_btax':'$'+str(tableprice),"invoice_no":"245421"})
    doc.save('new_invoice.docx')

def switch_buywin_specs(switch,page):
    a = [buywin_1_overview,buywin_1_feature,buywin_1_specification]
    c=[overviewcanvas,featurecanvas,specificationscanvas]
    x=0
    b=[[919.0,469.0],[1247.0,469.0],[1575.0,469.0]]
    for i in a :
    if i == switch:
    i.place(x=3000,y=3000)
    x+=1
    else:
    i.place(x=b[x][0],y=b[x][1])
    x+=1
    for k in c:
    if k==page:
    page.place(x=922.0,y=526.0)
    else:
    k.place(x=3000,y=3000)
```

```
def cur_sel_1():
        a = [brandcombo,kmsdrivencombo,model_yrcombo,modelcombo,rtocombo]
        state= True
        for i in a:
        if len(i.get()) < 2:
        state=False
        if state == True:
    slwin.place(x=3000,y=3000)
        sel_1.place(x=0,y=0)
        else:
        messagebox.showerror('Missing Data','Enter all the fields')
formatted_date =''
datetime_obj = ""
def sel_1_sel_2():
        global inputlist,datetime_obj,formatted_date
        varsl_1=[sl_1airbagscombo,sl_1drivetraincombo,sl_1enginetypecombo,sl_1fueltypecombo,sl_1historycom
bo,sl_1insurancecombo,sl_1kmdrivencombo,sl_1last_servicecombo,sl_1mileagecombo,sl_1ownercombo,sl_1steerin
gcombo,sl_1transmissiontypecombo,sl_1seatupholstrycombo,sl_1registeredincombo,sl_1registrationcombo]
        state = True
        for i in varsl_1:
        if i == sl_1registeredincombo:
        try:
        datetime_obj = datetime.strptime(sl_1registeredincombo.get(), "%d/%m/%Y")
        except ValueError:
        try:
                datetime_obj = datetime.strptime(sl_1registeredincombo.get(), "%d-%m-%Y")
        except ValueError:
                try:
                datetime_obj = datetime.strptime(sl_1registeredincombo.get(), "%d %m %Y")
                except ValueError:
                messagebox.showerror("Invalid date format","Please enter the date in dd/mm/yyyy format.")
                state = False
        try:
        formatted_date = datetime_obj.strftime("%Y-%m-%d %H:%M:%S")
        print(formatted_date )
        except:
        print('wth')
        if len(i.get()) == 0:
        state = False

        if state ==True:
        for i in varsl_1:
           inputlist.append(i.get())
        sel_1.place(x=3000,y=3000)
        sel_2.place(x=0,y=0)
        else:
        messagebox.showerror('Missing Data','Enter all the fields')

import math
import random


def pricecal():
        base_price = car_brands_prices.get(f'{brandcombo.get()}',900000)
```

```
          estimated_price = random.randint(base_price-201010,base_price+10010)

          if sl_1kmdrivencombo.get()=='Less than 10,000 km':
          estimated_price = base_price - math.ceil(random.randint(1000,8000) * 0.0046 )
   elif sl_1kmdrivencombo.get()== '10,000 - 25,000 km':
          estimated_price = base_price - math.ceil(random.randint(10000,20000)  * 0.0053 )

          elif sl_1kmdrivencombo.get()== '25,000 - 50,000 km':
          estimated_price = base_price - math.ceil(random.randint(25000,40000) * 0.0066 )

          elif sl_1kmdrivencombo.get()== '50,000 - 100,000 km':
          estimated_price = base_price - math.ceil(random.randint(50000,80000) * 0.0076 )

          elif sl_1kmdrivencombo.get()== 'More than 100,000 km':
          estimated_price = base_price - math.ceil(random.randint(100000,120000) * 0.009 )
          else:
          try:
          estimated_price = base_price - math.ceil(int(sl_1kmdrivencombo.get())*0.0439)
          except:
          estimated_price = base_price - 70000


   print(estimated_price)
          if sl_1ownercombo.get() =='First':
          estimated_price = estimated_price - 10000
          elif sl_1ownercombo.get() == 'Second':
          estimated_price = estimated_price - 20000
          elif sl_1ownercombo.get() == 'Third or more':
          estimated_price = estimated_price - 23000
          elif sl_1ownercombo.get() == 'Company owned':
          estimated_price = estimated_price
          else :
          estimated_price = estimated_price - 10000
          print(estimated_price)
          try:
          estimated_price =estimated_price - (int(datetime.now().year) - int(model_yrcombo.get() )) * 5000
          except:
          estimated_price = estimated_price - 40000
          print(estimated_price)
          return estimated_price

def sel_2_toselling():
          # get
          varsl_2=[orvmscombo,enginestart_topcombo,centrallockingcombo,sunroofcombo,rearaccombo,powerwind
owscombo,headlampscombo,maxpowercombo,fueltankcapcombo,seatingcapacitycombo,alterantefueltypecombo,mu
sicsystemcombo,transmissioncombo]
          state = True
          for i in varsl_2:
          if len(i.get()) == 0:
          state = False

          if state ==True:
          for i in varsl_2:
          inputlist.append(i.get())
          result = messagebox.askquestion('Data Confirmation','Are you sure the data you have entered is right')
          if result == 'yes':
```

```
    price= pricecal()
    result1 =messagebox.askquestion('Price Confirmation',f'The price of the car is Rs.{price}, are you sure you
want to sell it')
    if result1 == 'yes':
    uploadseldata()
    sel_2.place(x=3000)
    buywin.place(x=0,y=0)

    else:
    pass
    else:
    sel_1.place(x=0,y=0)
    sel_2.place(x=3000)

    else:
    messagebox.showerror('Missing Data','Enter all the fields')
file_paths=[]
def uploadpic(a):
    global file_paths
    file_paths1 = filedialog.askopenfilenames(filetypes=[("Image files", "*.jpg *.jpeg *.png")])
    print(len(file_paths1))
    if a == sel_2button_3:
    if (len(file_paths1)) <3:
    messagebox.showerror('Invalid Number of Pictures','Please upload minimum of three pictures')
    else:
    file_paths= file_paths1[0:3]
    if a == sel_2button_6:
    if (len(file_paths1)) <1:
    messagebox.showerror('Invalid Number of Pictures','Please upload minimum of one pictures')
    else: pass

def uploadseldata():
    connection = mysql.connector.connect(
    host='localhost',
    user='root',
    password='data@6420',
    database='cars'
)

    # Create a cursor to interact with the database
    cursor = connection.cursor()
    global file_paths
    # Read image data from files
    if file_paths:
    try:
    image_data = []

    # Read each image file as binary data
    for file_path in file_paths:
    with open(file_path, 'rb') as file:
        image_data.append(file.read())

    # Prepare the SQL query to insert the image data into the "pic" table
    pic_query = "INSERT INTO pic (pic1, pic2, pic3) VALUES (%s, %s, %s)"

    # Execute the query with the provided image data
```

```
        cursor.execute(pic_query, tuple(image_data))

         # Commit the changes to the database
        connection.commit()

        print("Images uploaded successfully!")
        except IOError as e:
        print("Error uploading images:", str(e))
        else:
        print("No images selected.")

        pic_id = cursor.lastrowid
        global datetime_obj,formatted_date


        column_names = ['idpic','price','brand','model','modelyr','History', 'kmsdriven', 'lastservice', 'registration',
'registered_in', 'owner',
                'fuel_type', 'transmission', 'insurance', 'airbags', 'seat_upholstery', 'music_system', 'ORVMs',
                'engine_start_stop', 'central_locking', 'sunroof', 'rear_ac', 'power_windows', 'headlamps',
'engine_type',
                'max_power', 'drivetrain', 'Fuel_capacity', 'mileage', 'seating_capacity', 'steering_type',
'alternate_fuel']
        varsl_2=[orvmscombo,enginestart_topcombo,centrallockingcombo,sunroofcombo,rearaccombo,powerwind
owscombo,headlampscombo,maxpowercombo,fueltankcapcombo,seatingcapacitycombo,alterantefueltypecombo,mu
sicsystemcombo,transmissioncombo]

        values = [pic_id,pricecal(),brandcombo.get(),modelcombo.get(),model_yrcombo.get(),
sl_1historycombo.get(), sl_1kmdrivencombo.get(), sl_1last_servicecombo.get(),
        sl_1registrationcombo.get(),formatted_date, sl_1ownercombo.get(),
sl_1fueltypecombo.get(),sl_1transmissiontypecombo.get(), sl_1insurancecombo.get(),
        sl_1airbagscombo.get(), sl_1seatupholstrycombo.get(),musicsystemcombo.get() ,orvmscombo.get(),
enginestart_topcombo.get(), centrallockingcombo.get(), sunroofcombo.get(), rearaccombo.get(),
powerwindowscombo.get(), headlampscombo.get(), sl_1enginetypecombo.get(),
        maxpowercombo.get(), sl_1drivetraincombo.get(), fueltankcapcombo.get(), sl_1mileagecombo.get(),
seatingcapacitycombo.get(),
        sl_1steeringcombo.get(), alterantefueltypecombo.get()]
        print(len(column_names))
        print(len(values))
        print(formatted_date )

        query = f"INSERT INTO details_of ({', '.join(column_names)}) VALUES ({', '.join(['%s'] *
len(column_names))})"
        cursor.execute(query, values)
        connection.commit()
def on_enter(event):
        event.widget['background'] = '#FCF7E4'
        event.widget['foreground'] = '#746C81'

def on_leave(event):
        event.widget['background'] = '#746C81'
        event.widget['foreground'] = '#FCF7E4'

def change_image_path(new_image_path):
        global picid_chosen
        cursor = mydb.cursor()
        sql = f"SELECT pic{new_image_path} FROM pic WHERE idpic = {picid_chosen}"
```

```
    mycursor.execute(sql)
    result = mycursor.fetchone()
    image = Image.open(io.BytesIO(result[0]))
    resized_image = image.resize((830, 400), resample=Image.LANCZOS)
    imgtk = ImageTk.PhotoImage(resized_image)
    #imgtk = ImageTk.PhotoImage(Image.open(io.BytesIO(result[0])).resize((830, 400)))
    buywin_1_veiw1_3.configure(image=imgtk)
    buywin_1_veiw1_3.image = imgtk


def import_files():
    progress["value"] += 33.33
    imp()
def update_data():
    progress["value"] -= 33.33

def imp():
    filetypes = (("Text files", "*.txt"), ("All files", "*.*"))
    files = filedialog.askopenfilenames(title="Select file(s)", filetypes=filetypes)
    for file in files:
    print("Imported file:", file)
def changebankchosen(tea):
    global bankchosen
    fin_b.itemconfigure(bankfirmb,text= tea)
    bankchosen = tea
def changebankchosen2(tea):
    global bankchosen
    fin_l.itemconfigure(bankfirml,text= tea)
    bankchosen=tea



#Master window
window = Tk()

window.geometry("1920x1080")
window.configure(bg = "#FFFFFF")


#opening window

opwin = Canvas(
    window,
    bg = "#FFFFFF",
    height = 1080,
    width = 1920,
    bd = 0,
    highlightthickness = 0,
    relief = "ridge"
)

opwin.place(x = 0, y = 0)
bg_opwin_img = PhotoImage(
    file=relative_to_assets("bg_opwindow.png"))
bg_opwin = opwin.create_image(
    965.0,
    540.0,
    image=bg_opwin_img
)

sgnup_opwin_img = PhotoImage(
    file=relative_to_assets("signup_opwindow.png"))
sgnup__opwin = Button(opwin,
    image=sgnup_opwin_img,
    borderwidth=0,
    highlightthickness=0,
    command=lambda: cur_sgnup(opwin),
    relief="flat"
)
sgnup__opwin.place(
    x=1332.6427001953125,
    y=685.2328491210938,
    width=283.714599609375,
    height=81.42266845703125
)

sgnin_opwin_img = PhotoImage(
    file=relative_to_assets("signin_opwindow.png"))
sgnin_opwin = Button(opwin,
    image=sgnin_opwin_img,
    borderwidth=0,
    highlightthickness=0,
    command=lambda: cur_sgnin(opwin),
    relief="flat"
)
sgnin_opwin.place(
    x=1332.6427001953125,
    y=452.7651672363281,
    width=283.714599609375,
    height=81.42269897460938
)

#sign in window

sgninwin = Canvas(
    window,
    bg = "#FFFFFF",
    height = 1080,
    width = 1920,
    bd = 0,
    highlightthickness = 0,
    relief = "ridge"
)

sgninwin.place(x = 1921, y = 0)
bg_sgninwin_img = PhotoImage(
    file=relative_to_assets("bg_sgninwin.png"))
bg_sgninwin_img_bg = sgninwin.create_image(
    960.0,
    540.0,
    image=bg_sgninwin_img
```

```
)

sgnin_sgninwin_img = PhotoImage(
        file=relative_to_assets("sgnin_sgninwin.png"))
sgnin_sgninwin = Button(sgninwin,
        image=sgnin_sgninwin_img,
        borderwidth=0,
        highlightthickness=0,
        command=lambda:check_user_credentials(username_sgninwin.get(), pass_sgninwin.get()),
        relief="flat"
)
sgnin_sgninwin.place(
        x=335.0,
        y=651.0,
        width=297.0,
        height=68.0
)

sgnup_sgninwin_img = PhotoImage(
        file=relative_to_assets("sgnup_sgninwin.png"))
sgnup_sgninwin = Button(sgninwin,
        image=sgnup_sgninwin_img,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: cur_sgnup(sgninwin),
        relief="flat"
)
sgnup_sgninwin.place(
        x=366.0,
        y=802.0,
        width=234.0,
        height=64.0
)

eye_sgninwin_img = PhotoImage(
        file=relative_to_assets("eye_sgninwin.png"))
eye_sgninwin = Button(sgninwin,
        image=eye_sgninwin_img,
        borderwidth=0,
        highlightthickness=0,
        command=lambda:reveal_password(pass_sgninwin,eye_sgninwin),
        relief="flat"
)
eye_sgninwin.place(
        x=788.0,
        y=541.0,
        width=48.0,
        height=62.0
)

pass_sgninwin_img = PhotoImage(
        file=relative_to_assets("pass_sgninwin.png"))
pass_sgninwin_img_bg = sgninwin.create_image(
        436.5,
        572.0,
        image=pass_sgninwin_img
```

```
)
pass_sgninwin = Entry(sgninwin,
            bd=0,
            bg="#FCF7E4",
            fg="#000716",
            highlightthickness=0,
            show="*"

)
pass_sgninwin.place(
            x=122.0,
            y=528.0,
            width=629.0,
            height=86.0
)

username_sgninwin_img = PhotoImage(
            file=relative_to_assets("username_sgninwin.png"))
username_sgninwin_img_bg = sgninwin.create_image(
            480.0,
            407.5,
            image=username_sgninwin_img
)
username_sgninwin = Entry(sgninwin,
            bd=0,
            bg="#FCF7E4",
            fg="#000716",
            highlightthickness=0
)
username_sgninwin.place(
            x=120.0,
            y=364.0,
            width=720.0,
            height=85.0
)


# signup

sgnupwin = Canvas(
            window,
            bg = "#FFFFFF",
            height = 1080,
            width = 1920,
            bd = 0,
            highlightthickness = 0,
            relief = "ridge"
)

sgnupwin.place(x = 3000, y = 3000)
bg_sgnupwin_img = PhotoImage(
            file=relative_to_assets("bg_sgnupwin.png"))
bg_sgnupwin = sgnupwin.create_image(
            960.0,
            540.0,
            image=bg_sgnupwin_img
```

```
)

bg_re_entr_sgnupwin_bg = PhotoImage(
        file=relative_to_assets("re_entr_sgnupwin.png"))
bg_re_entr_sgnupwin = sgnupwin.create_image(
        1428.0,
        681.5,
        image=bg_re_entr_sgnupwin_bg
)
re_entr_sgnupwin = Entry(sgnupwin,
        bd=0,
        bg="#FCF7E4",
        fg="#000716",
        highlightthickness=0,
        show="*"
)
re_entr_sgnupwin.place(
        x=1068.0,
        y=638.0,
        width=720.0,
        height=85.0
)

bg_username_sgnupwin_img = PhotoImage(
        file=relative_to_assets("bg_username_sgnupwin.png"))
username_sgnupwin_img = sgnupwin.create_image(
        1428.0,
        386.5,
        image=bg_username_sgnupwin_img
)
username_sgnupwin = Entry(sgnupwin,
        bd=0,
        bg="#FCF7E4",
        fg="#000716",
        highlightthickness=0
)
username_sgnupwin.place(
        x=1068.0,
        y=343.0,
        width=720.0,
        height=85.0
)

bg_pass_sgnupwin_img = PhotoImage(
        file=relative_to_assets("bg_pass_sgnupwin.png"))
pass_sgnupwin_img = sgnupwin.create_image(
        1428.0,
        532.5,
        image=bg_pass_sgnupwin_img
)
pass_sgnupwin = Entry(sgnupwin,
        bd=0,
        bg="#FCF7E4",
        fg="#000716",
        highlightthickness=0,
        show="*"
```

```
)
pass_sgnupwin.place(
	x=1068.0,
	y=484.0,
	width=720.0,
	height=95.0
)

eye_sgnupwin_img = PhotoImage(
	file=relative_to_assets("eye_sgnupwin.png"))
eye_sgnupwin = Button(sgnupwin,
	image=eye_sgnupwin_img,
	borderwidth=0,
	highlightthickness=0,
	command=lambda: reveal_password(pass_sgnupwin,eye_sgnupwin),
	relief="flat"
)
eye_sgnupwin.place(
	x=1748.0,
	y=508.0,
	width=48.0,
	height=48.0
)

sgnup_sgnupwin_img = PhotoImage(
	file=relative_to_assets("sgnin_sgnupwin.png"))
sgnup_sgnupwin = Button(sgnupwin,
	image=sgnup_sgnupwin_img,
	borderwidth=0,
	highlightthickness=0,
	command=lambda: cur_sgnin(sgnupwin),
	relief="flat"
)
sgnup_sgnupwin.place(
	x=1325.0,
	y=866.0,
	width=234.0,
	height=64.0
)

sgnin_sgnupwin_img = PhotoImage(
	file=relative_to_assets("sgnup_sgnupwin.png"))
sgnin_sgnupwin = Button(sgnupwin,
	image=sgnin_sgnupwin_img,
	borderwidth=0,
	highlightthickness=0,
	command=lambda: sign_func(username_sgnupwin.get(),pass_sgnupwin.get(),re_entr_sgnupwin.get()),
	relief="flat"
)
sgnin_sgnupwin.place(
	x=1292.0,
	y=748.0,
	width=297.0,
	height=68.0
)
```

```
#sell window



slwin = Canvas(
            window,
            bg = "#FFFFFF",
            height = 1080,
            width = 1920,
            bd = 0,
            highlightthickness = 0,
            relief = "ridge"
)

slwin.place(x = 3000, y = 3000)
bg_slwin_img = PhotoImage(
            file=relative_to_assets("bg_slwin.png"))
bg_slwin = slwin.create_image(
            961.0,
            540.0,
            image=bg_slwin_img
)

buy_slwin_img = PhotoImage(
            file=relative_to_assets("buy_slwin.png"))
buy_slwin = Button(slwin,
            image=buy_slwin_img,
            borderwidth=0,
            highlightthickness=0,
            command=lambda: cur_buy(slwin),
            relief="flat"
)
buy_slwin.place(
            x=913.0,
            y=84.0,
            width=205.0,
            height=53.0
)
brandcombo = ttk.Combobox(slwin,width=20, value = car_brands.get('brand') )
brandcombo.bind('<<ComboboxSelected>>', on_brandcombo_select)
brandcombo.place(x=585.0, y=474.0182800292969)

kmsdrivencombo = ttk.Combobox(slwin,width=20, value = car_brands.get('kmsdriven') )
kmsdrivencombo.place(x=585.0, y=561.2545471191406)

model_yrcombo = ttk.Combobox(slwin,width=20, value = car_brands.get('model_yr') )
model_yrcombo.place(x=585.0, y=648.4908142089844)

rtocombo = ttk.Combobox(slwin,width=20, value = car_brands.get('rto') )
rtocombo.place(x=585.0, y=735.7270812988281)

modelcombo = ttk.Combobox(slwin,width=20, value = [] )
modelcombo.place(x=585.0, y=822.9633483886719)
```

```
get_price_slwin_img= PhotoImage(
            file=relative_to_assets("get_price_slwin.png"))
get_price_slwin = Button(slwin,
            image=get_price_slwin_img,
            borderwidth=0,
            highlightthickness=0,
            command=lambda: cur_sel_1(),
            relief="flat"
)
get_price_slwin.place(
            x=1285.0,
            y=557.0,
            width=433.0,
            height=99.0
)

#buy window
buywin = Canvas(
            window,
    bg = "#FFFFFF",
            height = 1080,
            width = 1920,
            bd = 0,
            highlightthickness = 0,
            relief = "ridge"
)
buywin.place(x = 3000, y = 3000)
bg_buywin_img = PhotoImage(
            file=relative_to_assets("bg_buywin.png"))
bg_buywin = buywin.create_image(
            961.0,
            540.0,
            image=bg_buywin_img
)
x_coordinates = [198, 759, 1313]
y_coordinates = [656-75, 961-50]
abutton = tk.Button(buywin, text="Button 1", command=lambda: cur_buywin1(buywin_view1num), borderwidth=0,
                        activebackground="#FFF7D7", background="#746C81", bg="#746C81",

                        font=('Arial', 12, 'bold'))
abutton.place(x=x_coordinates[0], y=y_coordinates[0], width=300, height=50)
abutton.bind("<Enter>", on_enter)
abutton.bind("<Leave>", on_leave)

bbutton = tk.Button(buywin, text="Button 2", command=lambda: cur_buywin1(buywin_view3num), borderwidth=0,
                        activebackground="#FFF7D7", background="#746C81", bg="#746C81", foreground="#FCF7E4",
                        font=('Arial', 12, 'bold'))
bbutton.place(x=x_coordinates[1], y=y_coordinates[0], width=300, height=50)
bbutton.bind("<Enter>", on_enter)
bbutton.bind("<Leave>", on_leave)
cbutton = tk.Button(buywin, text="Button 3", command=lambda: cur_buywin1(buywin_view2num), borderwidth=0,
                         activebackground="#FFF7D7", background="#746C81", bg="#746C81",
foreground="#FCF7E4",
                        font=('Arial', 12, 'bold'))
cbutton.place(x=x_coordinates[2], y=y_coordinates[0], width=300, height=50)
cbutton.bind("<Enter>", on_enter)
cbutton.bind("<Leave>", on_leave)

dbutton = tk.Button(buywin, text="Button 4", command=lambda: cur_buywin1(buywin_view4num), borderwidth=0,
                    activebackground="#FFF7D7", background="#746C81", bg="#746C81", foreground="#FCF7E4",
                     font=('Arial', 12, 'bold'))
dbutton.place(x=x_coordinates[0], y=y_coordinates[1], width=300, height=50)
dbutton.bind("<Enter>", on_enter)
dbutton.bind("<Leave>", on_leave)

ebutton = tk.Button(buywin, text="Button 5", command=lambda: cur_buywin1(buywin_view6num), borderwidth=0,
                    activebackground="#FFF7D7", background="#746C81", bg="#746C81", foreground="#FCF7E4",
                    font=('Arial', 12, 'bold'))
ebutton.place(x=x_coordinates[1], y=y_coordinates[1], width=300, height=50)
ebutton.bind("<Enter>", on_enter)
ebutton.bind("<Leave>", on_leave)

fbutton = tk.Button(buywin, text="Button 6", command=lambda: cur_buywin1(buywin_view5num), borderwidth=0,
                    activebackground="#FFF7D7", background="#746C81", bg="#746C81", foreground="#FCF7E4",
                    font=('Arial', 12, 'bold'))
fbutton.place(x=x_coordinates[2], y=y_coordinates[1], width=300, height=50)
fbutton.bind("<Enter>", on_enter)
fbutton.bind("<Leave>", on_leave)

bbrandcombo = ttk.Combobox(buywin,width=20, value = car_brands.get('brand') )
bbrandcombo.set("Brand")
bbrandcombo.bind('<<ComboboxSelected>>', on_bbrandcombo_select)
bbrandcombo.place(x=289, y=300)

bkmsdrivencombo = ttk.Combobox(buywin,width=20, value = car_brands.get('kmsdriven') )
bkmsdrivencombo.set("Km driven")
bkmsdrivencombo.place(x=526.0, y=300)

bmodel_yrcombo = ttk.Combobox(buywin,width=20, value = car_brands.get('model_yr') )
bmodel_yrcombo.set("Model year")
bmodel_yrcombo.place(x=763.0, y=300)

brtocombo = ttk.Combobox(buywin,width=20, value = car_brands.get('rto') )
brtocombo.set("Registration")
brtocombo.place(x=1000.0, y=300)

bmodelcombo = ttk.Combobox(buywin,width=20, value = [] )
bmodelcombo.set("Model")
bmodelcombo.place(x=1237.0, y=300)
bg_srchbar_buywin_img = PhotoImage(
        file=relative_to_assets("entry_1.png"))
srchbar_buywin_img = buywin.create_image(
        869.0,
        237.0,
        image=bg_srchbar_buywin_img
)
srchbar_buywin = Entry(buywin,
        bd=0,
        bg="#9E99A7",
        fg="#000716",
        highlightthickness=0
)
```

```
srchbar_buywin.place(
    x=334.0,
    y=210.0,
    width=1070.0,
    height=52.0
)

srch_btn_img = PhotoImage(
    file=relative_to_assets("srch_btn_img.png"))
srch_btn = Button(buywin,
    image=srch_btn_img,
    borderwidth=0,
    highlightthickness=0,
    command=lambda: print("search clicked"),
    relief="flat"
)
srch_btn.place(
    x=1474.0,
    y=296.0,
    width=38.0,
    height=38.0
)
srch_btn_img2 = PhotoImage(
    file=relative_to_assets("srch_btn_img.png"))
srch2_btn_img = Button(buywin,
    image=srch_btn_img2,
    borderwidth=0,
    highlightthickness=0,
    command=lambda: print("search clicked"),
    relief="flat"
)
srch2_btn_img.place(
    x=1419.0,
    y=216.0,
    width=38.0,
    height=38.0
)
sell_buywin_img = PhotoImage(
    file=relative_to_assets("sell_buywin_img.png"))
sell_buywin = Button(buywin,
    image=sell_buywin_img,
    borderwidth=0,
    highlightthickness=0,
    command=lambda: cur_slwin(buywin),
    relief="flat"
)
sell_buywin.place(
    x=973.0,
    y=75.0,
    width=205.0,
    height=53.0
)

try:
    global mydb
    mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="data@6420",
    database="cars"
    )
except Error as e:
    print("Error connecting to MySQL database: {}".format(e))


mycursor = mydb.cursor()


idpic = 2
try:
    sql = "SELECT pic1, pic2, pic3 FROM pic WHERE idpic = %s"
    values = (idpic,)
    mycursor.execute(sql, values)
    result = mycursor.fetchone()
except mysql.connector.Error as e:
    print("Error retrieving images: {}".format(e))



photo1 = ImageTk.PhotoImage(Image.open(io.BytesIO(result[0])).resize((457, 220)))
photo2 = ImageTk.PhotoImage(Image.open(io.BytesIO(result[1])).resize((226, 117)))
photo3 = ImageTk.PhotoImage(Image.open(io.BytesIO(result[2])).resize((226, 117)))

buywin_button_image_9 = PhotoImage(
    file=relative_to_assets("button_9.png"))
buywin_view1 = Button(buywin,
    image=photo1,
    borderwidth=0,
    highlightthickness=0,
    command=lambda: cur_buywin1(buywin_view1num),
    relief="flat"
)

buywin_view1.place(
    x=137.0,
    y=422.0-75,
    width=457.0,
    height=220.0
)

buywin_button_image_10 = PhotoImage(
    file=relative_to_assets("button_10.png"))
buywin_view2 = Button(buywin,
    image=photo2,
    borderwidth=0,
    highlightthickness=0,
    command=lambda: cur_buywin1(buywin_view2num),
    relief="flat"
)
buywin_view2.place(
    x=1252.0,
    y=422.0-75,
    width=420.0,
    height=220.0
)

buywin_button_image_11 = PhotoImage(
    file=relative_to_assets("button_11.png"))
buywin_view3 = Button(buywin,
    image=photo3,
    borderwidth=0,
    highlightthickness=0,
    command=lambda: cur_buywin1(buywin_view3num),
    relief="flat"
)
buywin_view3.place(
    x=697.0,
    y=422.0-75,
    width=429.0,
    height=220.0
)

buywin_button_image_12 = PhotoImage(
    file=relative_to_assets("button_12.png"))
buywin_view4 = Button(buywin,
    image=photo2,
    borderwidth=0,
    highlightthickness=0,
    command=lambda: cur_buywin1(buywin_view4num),
    relief="flat"
)
buywin_view4.place(
    x=136.0,
    y=752.0-75,
    width=457.0,
    height=220.0
)

buywin_button_image_13 = PhotoImage(
    file=relative_to_assets("button_13.png"))
buywin_view5 = Button(buywin,
    image=photo1,
    borderwidth=0,
    highlightthickness=0,
    command=lambda: cur_buywin1(buywin_view5num),
    relief="flat"
)
buywin_view5.place(
    x=1251.0,
    y=751.0-75,
    width=420.0,
    height=220.0
)

button_image_14 = PhotoImage(
    file=relative_to_assets("button_14.png"))
buywin_view6 = Button(buywin,
    image=photo1,
    borderwidth=0,
    highlightthickness=0,
    command=lambda: cur_buywin1(buywin_view6num),
    relief="flat"
)
buywin_view6.place(
    x=696.0,
    y=751.0-75,
    width=429.0,
    height=220.0
)


switch_buywin_imgg = PhotoImage(
    file=relative_to_assets("switch_buywin.png"))
switch_buywin_img = buywin.create_image(
    881,
    1040,
    image=switch_buywin_imgg
)


switch_buywin_imgg = PhotoImage(
    file=relative_to_assets("switch1_buywin.png"))
switch1_buywin = Button(buywin,
    image=switch_buywin_imgg,
    borderwidth=0,
    highlightthickness=0,
    command=lambda: switch_sel(switch1_buywin),
    relief="flat"
)
switch1_buywin.place(
    x=3000,
    y=1040,
    width=16,
    height=16
)
switch2_buywin = Button(buywin,
    image=switch_buywin_imgg,
    borderwidth=0,
    highlightthickness=0,
    command=lambda: switch_sel(switch2_buywin),
    relief="flat"
)
switch2_buywin.place(
    x=945,
    y=1040,
    width=16,
    height=16
)

switch3_buywin = Button(buywin,
    image=switch_buywin_imgg,
    borderwidth=0,
    highlightthickness=0,
    command=lambda: switch_sel(switch3_buywin),
		relief="flat"
)
switch3_buywin.place(
		x=1009,
		y=1040,
		width=16,
		height=16
)




registration_no="0"
model_name="Honda Civic"
driven = "10000"
price="4,00,000"
fuel_type="petrol"
down_payement="0000"

output1="hello"
output2="hello"
output3="hello"
output4="hello"
output5="hello"
output6="hello"
output7="hello"
output8="hello"
output9="hello"
output10="hello"


list1= ["",1,2,3,4,5,6,7,8,9,10]
buywin_1 = Canvas(
		window,
		bg = "#FFFFFF",
		height = 1080,
		width = 1920,
		bd = 0,
		highlightthickness = 0,
		relief = "ridge"
)

buywin_1.place(x = 3000, y = 3000)
buywin_1_bg_img = PhotoImage(
		file=relative_to_assets("buywin_1_bg.png"))
buywin_1_bg = buywin_1.create_image(
		961.0,
		540.0,
		image=buywin_1_bg_img
)

buywin_1_button_image_1 = PhotoImage(
		file=relative_to_assets("buywin_1_buy.png"))
buywin_1_buy_1 = Button(buywin_1,
		image=buywin_1_button_image_1,
        borderwidth=0,
        highlightthickness=0,
        command=lambda:cur_buy(buywin_1),
        relief="flat"
)
buywin_1_buy_1.place(
        x=913.0,
        y=84.0,
        width=205.0,
        height=53.0
)

buywin_1_button_image_2 = PhotoImage(
        file=relative_to_assets("buywin_1_sell.png"))
buywin_1_sell_2 = Button(buywin_1,
        image=buywin_1_button_image_2,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: cur_slwin(buywin_1),
        relief="flat"
)
buywin_1_sell_2.place(
        x=1213.0,
        y=84.0,
        width=205.0,
        height=53.0
)
photo4 = ImageTk.PhotoImage(Image.open(io.BytesIO(result[0])).resize((830, 400)))
photo4_1 = ImageTk.PhotoImage(Image.open(io.BytesIO(result[0])).resize((226, 117)))
photo5= ImageTk.PhotoImage(Image.open(io.BytesIO(result[1])).resize((226, 117)))
photo6 = ImageTk.PhotoImage(Image.open(io.BytesIO(result[2])).resize((226, 117)))
buywin_1_button_image_3 = PhotoImage(
        file=relative_to_assets("buywin_1_veiw1.png"))
buywin_1_veiw1_3 = Button(buywin_1,
        image=photo4,
        borderwidth=0,
        highlightthickness=0,

        relief="flat"
)
buywin_1_veiw1_3.place(
        x=51.0,
        y=390.0,
        width=830.0,
        height=400.0
)

buywin_1_button_image_4 = PhotoImage(
        file=relative_to_assets("buywin_1_veiw2.png"))
buywin_1_veiw2_4 = Button(buywin_1,
        image=photo4_1,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: change_image_path(1),
        relief="flat"
)
```

```
buywin_1_veiw2_4.place(
        x=84.0,
        y=816.0,
        width=226.0,
        height=117.0
)

buywin_1_button_image_5 = PhotoImage(
        file=relative_to_assets("buywin_1_veiw3.png"))
buywin_1_veiw3_5 = Button(buywin_1,
        image=photo5,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: change_image_path(2),
        relief="flat"
)
buywin_1_veiw3_5.place(
        x=622.0,
        y=816.0,
        width=226.0,
        height=117.0
)

buywin_1_button_image_6 = PhotoImage(
        file=relative_to_assets("buywin_1_veiw4.png"))
buywin_1_veiw4_6 = Button(buywin_1,
        image=photo6,
        borderwidth=0,
        highlightthickness=0,
  command=lambda: change_image_path(3),
        relief="flat"
)
buywin_1_veiw4_6.place(
        x=353.0,
        y=816.0,
        width=226.0,
        height=117.0
)

buywin_1_image_2 = PhotoImage(
        file=relative_to_assets("buywin_1_topbar.png"))
buywin_1_topbar_image_2 = buywin_1.create_image(
        984.0,
        274.0,
        image=buywin_1_image_2
)

buy_c_name = buywin_1.create_text(
        360.0,
        240.0,
        anchor="nw",
        text=model_name,
        fill="#FFF7D7",
        font=("Inter Medium", 30 * -1)
)
```

```
buy_c_kms=buywin_1.create_text(
        369.0,
        278.0,
        anchor="nw",
        text=str(driven)+"km",
        fill="#FFF7D7",
        font=("Inter Medium", 20 * -1)
)

buy_c_fuel_t =buywin_1.create_text(
        492.0+30,
        278.0,
        anchor="nw",
        text=fuel_type,
        fill="#FFF7D7",
        font=("Inter Medium", 20 * -1)
)

buy_c_regis =buywin_1.create_text(
        591.0,
        279.0,
        anchor="nw",
        text=registration_no,
        fill="#FFF7D7",
        font=("Inter Medium", 20 * -1)
)
buywin_1_button_image_7 = PhotoImage(
        file=relative_to_assets("buywin_1_buynow.png"))
buywin_1_buynow_button_7 = Button(buywin_1,
        image=buywin_1_button_image_7,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: cur_buynow(buywin_1),
        relief="flat"
)
buywin_1_buynow_button_7.place(
        x=1133.0,
        y=244.0,
        width=266.0,
        height=64.0
)

buywin_1_button_image_8 = PhotoImage(
        file=relative_to_assets("buywin_1_testdrive.png"))
buywin_1_testdrive_button_8 = Button(buywin_1,
        image=buywin_1_button_image_8,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: print("button_8 clicked"),
        relief="flat"
)
buywin_1_testdrive_button_8.place(
        x=1424.0,
        y=244.0,
        width=266.0,
        height=64.0
```

```
)

buywin_1_button_image_9 = PhotoImage(
        file=relative_to_assets("buywin_1_back.png"))
buywin_1_back_button_9 = Button(buywin_1,
        image=buywin_1_button_image_9,
        borderwidth=0,
        highlightthickness=0,
        command=lambda:cur_buy(buywin_1),
        relief="flat"
)
buywin_1_back_button_9.place(
        x=274.0,
        y=255.0,
        width=25.0,
        height=38.0
)

buywin_1.create_rectangle(
        84.0,
        949.0,
        310.0,
        949.0,
        fill="#FFFFFF",
        outline="")

buywin_1.create_rectangle(
        622.0,
        949.0,
        848.0,
        949.0,
        fill="#FFFFFF",
        outline="")

buywin_1.create_rectangle(
        353.0,
        949.0,
        579.0,
        949.0,
        fill="#FFFFFF",
        outline="")

buy_c_price=buywin_1.create_text(
        914.0,
        244.0,
        anchor="nw",
        text=price,
        fill="#FFF7D7",
        font=("Inter Medium", 19 * -1)
)

buy_c_downpayment = buywin_1.create_text(
        914.0,
        276.0,
        anchor="nw",
        text=down_payement,
        fill="#FFF7D7",
        font=("Inter Medium", 19 * -1)
)

eh = -3


#overview


overviewcanvas = Canvas(
        buywin_1,
        bg = "#FFFFFF",
        height = 325,
        width = 973,
        bd = 0,
        highlightthickness = 0,
        relief = "ridge"
)

overviewcanvas.place(x = 922, y = 526)
ovrcaimage_image_1 = PhotoImage(
        file=relative_to_assets("overviewca.png"))
ovrcaimage_1 = overviewcanvas.create_image(
        486.0,
        162.0,
        image=ovrcaimage_image_1
)

bhistory = overviewcanvas.create_text(
        285.0,
        32.0+eh,
        anchor="nw",
        text=tableHistory,
        fill="#FFF7D7",
        font=("Inter Medium", 21 * -1)
)

bkms=overviewcanvas.create_text(
        284.0,
        90.0+eh,
        anchor="nw",
        text="2",
        fill="#FFF7D7",
        font=("Inter Medium", 21 * -1)
)

blst_service =overviewcanvas.create_text(
        284.0,
        146.0+eh,
        anchor="nw",
        text="3",
        fill="#FFF7D7",
        font=("Inter Medium", 21 * -1)
)
```

```
bregistration= overviewcanvas.create_text(
        285.0,
        209.0+eh,
        anchor="nw",
        text="4",
        fill="#FFF7D7",
        font=("Inter Medium", 21 * -1)
)

bregisteredin = overviewcanvas.create_text(
        285.0,
        271.0+eh,
        anchor="nw",
        text="5",
        fill="#FFF7D7",
  font=("Inter Medium", 21 * -1)
)
bowner=overviewcanvas.create_text(
        754.0,
        32.0+eh,
        anchor="nw",
        text="6",
        fill="#FFF7D7",
        font=("Inter Medium", 21 * -1)
)

bfuel_type= overviewcanvas.create_text(
        754.0,
        92.0+eh,
        anchor="nw",
        text="7",
        fill="#FFF7D7",
        font=("Inter Medium", 21 * -1)
)

btransmission = overviewcanvas.create_text(
        754.0,
        144.0+eh,
        anchor="nw",
        text="8",
        fill="#FFF7D7",
        font=("Inter Medium", 21 * -1)
)

binsurance=overviewcanvas.create_text(
        754.0,
        209.0+eh,
        anchor="nw",
        text="9",
        fill="#FFF7D7",
        font=("Inter Medium", 21 * -1)
)


featurecanvas = Canvas(
        buywin_1,
        bg = "#FFFFFF",
        height = 325,
        width = 973,
        bd = 0,
        highlightthickness = 0,
        relief = "ridge"
)

featurecanvas.place(x = 9190, y = 5230)
ftcaimage_image_1 = PhotoImage(
        file=relative_to_assets("featureca.png"))
ftcaimage_1 = featurecanvas.create_image(
        486.0,
        162.0,
        image=ftcaimage_image_1
)

fairbags =featurecanvas.create_text(
        285.0,
        32.0+eh,
        anchor="nw",
        text="1",
        fill="#FFF7D7",
        font=("Inter Medium", 21 * -1)
)

fseatupholstry =featurecanvas.create_text(
        284.0,
        90.0+eh,
        anchor="nw",
        text="2",
        fill="#FFF7D7",
        font=("Inter Medium", 21 * -1)
)

fmusicsys =featurecanvas.create_text(
        284.0,
        146.0+eh,
        anchor="nw",
        text="3",
        fill="#FFF7D7",
        font=("Inter Medium", 21 * -1)
)

fovrm =featurecanvas.create_text(
        285.0,
        209.0+eh,
        anchor="nw",
        text="4",
        fill="#FFF7D7",
        font=("Inter Medium", 21 * -1)
)

fenginestrt =featurecanvas.create_text(
        285.0,
        271.0+eh,
		anchor="nw",
		text="5",
		fill="#FFF7D7",
		font=("Inter Medium", 21 * -1)
)
fcentralock =featurecanvas.create_text(
		754.0,
		32.0+eh,
		anchor="nw",
		text="6",
		fill="#FFF7D7",
		font=("Inter Medium", 21 * -1)
)

fsunroof =featurecanvas.create_text(
		754.0,
		92.0+eh,
		anchor="nw",
		text="7",
		fill="#FFF7D7",
		font=("Inter Medium", 21 * -1)
)

frearac =featurecanvas.create_text(
		754.0,
		144.0+eh,
		anchor="nw",
		text="8",
		fill="#FFF7D7",
		font=("Inter Medium", 21 * -1)
)

fpowrwind =featurecanvas.create_text(
		754.0,
		209.0+eh,
		anchor="nw",
		text="9",
		fill="#FFF7D7",
		font=("Inter Medium", 21 * -1)
)

fheadlamps =featurecanvas.create_text(
		757.0,
		271.0+eh,
		anchor="nw",
		text="10",
		fill="#FFF7D7",
		font=("Inter Medium", 21 * -1)
)
specificationscanvas = Canvas(
		buywin_1,
		bg = "#FFFFFF",
		height = 325,
		width = 973,
		bd = 0,
		highlightthickness = 0,
            relief = "ridge"
)

specificationscanvas.place(x = 9220, y = 5260)
speccaimage_image_1 = PhotoImage(
            file=relative_to_assets("specificationsca.png"))
speccaimage_1 = specificationscanvas.create_image(
            486.0,
            162.0,
            image=speccaimage_image_1
)

specfuel_type =specificationscanvas.create_text(
            285.0,
            32.0+eh,
            anchor="nw",
            text="1",
            fill="#FFF7D7",
            font=("Inter Medium", 21 * -1)
)

specengine_type =specificationscanvas.create_text(
            284.0,
            90.0+eh,
            anchor="nw",
            text="2",
            fill="#FFF7D7",
            font=("Inter Medium", 21 * -1)
)

specdrivetrain =specificationscanvas.create_text(
            284.0,
    146.0+eh,
            anchor="nw",
            text="3",
            fill="#FFF7D7",
            font=("Inter Medium", 21 * -1)
)

specmileage =specificationscanvas.create_text(
            285.0,
            209.0+eh,
            anchor="nw",
            text="4",
            fill="#FFF7D7",
            font=("Inter Medium", 21 * -1)
)

specsteeringtyp =specificationscanvas.create_text(
            285.0,
            271.0+eh,
            anchor="nw",
            text="5",
            fill="#FFF7D7",
            font=("Inter Medium", 21 * -1)
)
```

```
spectransmissiontyp =specificationscanvas.create_text(
        754.0,
        32.0+eh,
        anchor="nw",
        text="6",
        fill="#FFF7D7",
        font=("Inter Medium", 21 * -1)
)

specmaxpwr =specificationscanvas.create_text(
        754.0,
        92.0+eh,
        anchor="nw",
        text="7",
        fill="#FFF7D7",
        font=("Inter Medium", 21 * -1)
)

specfueltank =specificationscanvas.create_text(
        754.0,
        144.0+eh,
        anchor="nw",
        text="8",
        fill="#FFF7D7",
        font=("Inter Medium", 21 * -1)
)

specseatincap =specificationscanvas.create_text(
        754.0,
        209.0+eh,
        anchor="nw",
        text="9",
        fill="#FFF7D7",
        font=("Inter Medium", 21 * -1)
)

speclternatefueltype =specificationscanvas.create_text(
        757.0,
        271.0+eh,
        anchor="nw",
        text="10",
        fill="#FFF7D7",
        font=("Inter Medium", 21 * -1)
)
#########################################################################################
##################



buywin_1_button_image_10 = PhotoImage(
        file=relative_to_assets("buywin_1_overview_c.png"))
buywin_1_overview_c = Button(buywin_1,
        image=buywin_1_button_image_10,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: print("button_10 clicked"),
        relief="flat"
)
buywin_1_overview_c.place(
        x=919.0,
        y=469.0,
        width=323.3333435058594,
        height=37.0
)

buywin_1_button_image_11 = PhotoImage(
        file=relative_to_assets("buywin_1_feature_c.png"))
buywin_1_feature_c = Button(buywin_1,
        image=buywin_1_button_image_11,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: print("button_11 clicked"),
        relief="flat"
)
buywin_1_feature_c.place(
        x=1247.0,
        y=469.0,
        width=323.3333740234375,
        height=37.0
)

buywin_1_button_image_12 = PhotoImage(
        file=relative_to_assets("buywin_1_specification_c.png"))
buywin_1_specification_c = Button(buywin_1,
        image=buywin_1_button_image_12,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: print("button_12 clicked"),
        relief="flat"
)
buywin_1_specification_c.place(
        x=1575.0,
        y=469.0,
        width=323.3333740234375,
        height=37.0
)

buywin_1_button_image_13 = PhotoImage(
        file=relative_to_assets("buywin_1_overview.png"))
buywin_1_overview = Button(buywin_1,
        image=buywin_1_button_image_13,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: switch_buywin_specs(buywin_1_overview,overviewcanvas),
        relief="flat"
)
buywin_1_overview.place(
        x=3000,
        y=469.0,
        width=323.3333435058594,
        height=37.0
)
```

```
buywin_1_button_image_14 = PhotoImage(
        file=relative_to_assets("buywin_1_feature.png"))
buywin_1_feature = Button(buywin_1,
        image=buywin_1_button_image_14,
        borderwidth=0,
        highlightthickness=0,
        command=lambda:switch_buywin_specs(buywin_1_feature,featurecanvas),
        relief="flat"
)
buywin_1_feature.place(
        x=1247.0,
        y=469.0,
        width=323.3333740234375,
        height=37.0
)

buywin_1_button_image_15 = PhotoImage(
        file=relative_to_assets("buywin_1_specification.png"))
buywin_1_specification = Button(buywin_1,
        image=buywin_1_button_image_15,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: switch_buywin_specs(buywin_1_specification,specificationscanvas),
        relief="flat"
)
buywin_1_specification.place(
        x=1575.0,
        y=469.0,
        width=323.3333740234375,
        height=37.0
)

#fin_select
fin_s = Canvas(
        window,
        bg = "#FFFFFF",
        height = 1080,
        width = 1920,
        bd = 0,
        highlightthickness = 0,
        relief = "ridge"
)

fin_s.place(x = 3000, y = 0)
fin_simage_image_1 = PhotoImage(
        file=relative_to_assets("fin_simage_1.png"))
fin_simage_1 = fin_s.create_image(
        960.0,
        540.0,
        image=fin_simage_image_1
)

fin_sbutton_image_1 = PhotoImage(
        file=relative_to_assets("fin_sbutton_1.png"))
fin_sbutton_1 = Button(fin_s,
        image=fin_sbutton_image_1,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: cur_buy(fin_s),
        relief="flat"
)
fin_sbutton_1.place(
        x=913.5242309570312,
        y=84.0,
        width=204.89328002929688,
        height=53.0
)

fin_sbutton_image_2 = PhotoImage(
        file=relative_to_assets("fin_sbutton_2.png"))
fin_sbutton_2 = Button(fin_s,
        image=fin_sbutton_image_2,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: fin_s_respect(fin_b),
        relief="flat"
)
fin_sbutton_2.place(
        x=1225.0,
        y=586.0,
        width=412.0,
        height=86.0
)

fin_sbutton_image_3 = PhotoImage(
        file=relative_to_assets("fin_sbutton_3.png"))
fin_sbutton_3 = Button(fin_s,
        image=fin_sbutton_image_3,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: fin_s_respect(fin_l),
        relief="flat"
)
fin_sbutton_3.place(
        x=1225.0,
        y=764.0,
  width=412.0,
        height=86.0
)

fin_sbutton_image_4 = PhotoImage(
        file=relative_to_assets("fin_sbutton_4.png"))
fin_sbutton_4 = Button(fin_s,
        image=fin_sbutton_image_4,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: cur_slwin(fin_s),
        relief="flat"
)
fin_sbutton_4.place(
        x=1213.0,
        y=84.0,
        width=205.0,
        height=53.0
)

car_namef =fin_s.create_text(
        140.0,
        352.0,
        anchor="nw",
        text="Car: Honda Civic 2017",
        fill="#746C81",
        font=("Inter Bold", 35* -1)
)

car_namef2 =fin_s.create_text(
        1255.0,
        373.0,
        anchor="nw",
        text="Honda Civic",
        fill="#746C81",
        font=("Inter Medium", 45 * -1)
)


photo10 = ImageTk.PhotoImage(Image.open(io.BytesIO(result[0])).resize((950, 520)))
fin_sbutton_image_5 = PhotoImage(
        file=relative_to_assets("fin_sbutton_5.png"))
fin_sbutton_5 = Button(fin_s,
        image=photo10,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: print("nothin"),
        relief="flat"
)
fin_sbutton_5.place(
        x=140.0,
        y=400.0,
        width=950.0,
        height=520.0
)

fin_s.create_text(
        757.0,
        212.0,
        anchor="nw",
        text="Financing Options",
        fill="#746C81",
        font=("Inter SemiBold", 50 * -1)
)

fin_s.create_text(
        1320.0,
        537.0,
        anchor="nw",
        text="Full payment?",
        fill="#746C81",
        font=("Inter", 34 * -1)
)

fin_s.create_text(
        1235.0,
        432.0,
        anchor="nw",
        text="Price:",
        fill="#746C81",
        font=("Inter", 48 * -1)
)

car_pricef=fin_s.create_text(
        1370.0,
  432.0,
        anchor="nw",
        text="₹4,00,000",
        fill="#746C81",
        font=("Inter Medium", 48 * -1)
)

fin_s.create_text(
        1307.0,
        710.0,
        anchor="nw",
        text="Through Loan?",
        fill="#746C81",
        font=("Inter", 35 * -1)
)


######### fin_b


fin_b = Canvas(
        window,
  bg = "#FFFFFF",
        height = 1080,
        width = 1920,
        bd = 0,
        highlightthickness = 0,
        relief = "ridge"
)

fin_b.place(x = 30000, y = 0)
fin_bimage_image_1 = PhotoImage(
        file=relative_to_assets("fin_bimage_1.png"))
fin_bimage_1 = fin_b.create_image(
        960.0,
        540.0,
        image=fin_bimage_image_1
)

fin_bbutton_image_1 = PhotoImage(
        file=relative_to_assets("fin_bbutton_1.png"))
fin_bbutton_1 = Button(fin_b,
        image=fin_bbutton_image_1,
        borderwidth=0,
        highlightthickness=0,
  command=lambda: cur_buy(fin_b),
        relief="flat"
)
fin_bbutton_1.place(
        x=913.5242309570312,
        y=84.0,
        width=204.89328002929688,
        height=53.0
)

fin_bbutton_image_2 = PhotoImage(
        file=relative_to_assets("fin_bbutton_2.png"))
fin_bbutton_2 = Button(fin_b,
        image=fin_bbutton_image_2,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: cur_slwin(fin_b),
        relief="flat"
)
fin_bbutton_2.place(
        x=1213.0,
        y=84.0,
        width=205.0,
        height=53.0
)



price_nd_nameb =fin_b.create_text(
        242.0,
        577.0,
        anchor="nw",
        text="₹10,00,000",
        fill="#746C81",
        font=("Inter", 34 * -1)
)


fin_bbutton_image_3 = PhotoImage(
        file=relative_to_assets("fin_bbutton_3.png"))
fin_bbutton_3 = Button(fin_b,
        image=fin_bbutton_image_3,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: changebankchosen("State Bank of India"),
        relief="flat"
)
fin_bbutton_3.place(
        x=133.0,
        y=376.0,
        width=225.0,
        height=94.0
)
```

```
fin_bbutton_image_4 = PhotoImage(
        file=relative_to_assets("fin_bbutton_4.png"))
fin_bbutton_4 = Button(fin_b,
        image=fin_bbutton_image_4,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: changebankchosen("South Indian Bank"),
        relief="flat"
)
fin_bbutton_4.place(
        x=381.0,
        y=376.0,
        width=225.0,
        height=94.0
)

fin_bbutton_image_5 = PhotoImage(
        file=relative_to_assets("fin_bbutton_5.png"))
fin_bbutton_5 = Button(fin_b,
        image=fin_bbutton_image_5,
        borderwidth=0,
        highlightthickness=0,
        command=lambda:changebankchosen("Axis Bank"),
        relief="flat"
)
fin_bbutton_5.place(
        x=629.0,
        y=376.0,
        width=225.0,
        height=94.0
)

fin_bbutton_image_6 = PhotoImage(
        file=relative_to_assets("fin_bbutton_6.png"))
fin_bbutton_6 = Button(fin_b,
        image=fin_bbutton_image_6,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: changebankchosen("HDFC Bank"),
        relief="flat"
)
fin_bbutton_6.place(
        x=877.0,
        y=376.0,
        width=225.0,
        height=94.0
)

fin_bbutton_image_7 = PhotoImage(
        file=relative_to_assets("fin_bbutton_7.png"))
fin_bbutton_7 = Button(fin_b,
        image=fin_bbutton_image_7,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: update_data(),
        relief="flat"
)
fin_bbutton_7.place(
  x=725.0,
        y=813.0,
        width=30.08770751953125,
        height=35.0
)

fin_bbutton_image_8 = PhotoImage(
        file=relative_to_assets("fin_bbutton_8.png"))
fin_bbutton_8 = Button(fin_b,
        image=fin_bbutton_image_8,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: import_files(),
        relief="flat"
)
fin_bbutton_8.place(
        x=607.0,
        y=722.0,
        width=99.0,
        height=81.0
)

fin_bbutton_image_9 = PhotoImage(
        file=relative_to_assets("fin_bbutton_9.png"))
fin_bbutton_9 = Button(fin_b,
        image=fin_bbutton_image_9,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: import_files(),
        relief="flat"
)
fin_bbutton_9.place(
        x=1587.0,
        y=722.0,
        width=99.0,
        height=81.0
)

fin_bbutton_image_10 = PhotoImage(
        file=relative_to_assets("fin_bbutton_10.png"))
fin_bbutton_10 = Button(fin_b,
        image=fin_bbutton_image_10,
        borderwidth=0,
        highlightthickness=0,
        command=lambda:import_files(),
        relief="flat"
)
fin_bbutton_10.place(
        x=1058.0,
        y=722.0,
        width=99.0,
        height=81.0
)
```

```
entry_image_1 = PhotoImage(
            file=relative_to_assets("fin_bentry_1.png"))
entry_bg_1 = fin_b.create_image(
            809.5,
            535.0,
            image=entry_image_1
)
entry_1 = Entry(fin_b,
            bd=0,
            bg="#FFF7D7",
            fg="#000716",
            highlightthickness=0
)
entry_1.place(
            x=520.0,
            y=512.0,
            width=579.0,
            height=44.0
)

bankfirmb = fin_b.create_text(
            362.0,
            308.0,
            anchor="nw",
            text="Choose a bank",
            fill="#746C81",
            font=("Inter", 32 * -1)
)

fin_bbutton_image_11 = PhotoImage(
            file=relative_to_assets("fin_bbutton_11.png"))
fin_bbutton_11 = Button(fin_b,
            image=fin_bbutton_image_11,
            borderwidth=0,
            highlightthickness=0,
            command=lambda: cur_invoice(fin_b),
            relief="flat"
)
fin_bbutton_11.place(
            x=733.0,
            y=905.0,
            width=363.0,
            height=71.0
)

fin_bbutton_image_12 = PhotoImage(
            file=relative_to_assets("fin_bbutton_12.png"))
fin_bbutton_12 = Button(fin_b,
            image=fin_bbutton_image_12,
            borderwidth=0,
   highlightthickness=0,
            command=lambda: print("button_12 clicked"),
            relief="flat"
)
fin_bbutton_12.place(
            x=575.0,
            y=813.0,
            width=147.0,
            height=34.0
)


fin_bbutton_13 = Button(fin_b,
            image=fin_bbutton_image_7,
            borderwidth=0,
            highlightthickness=0,
            command=lambda: update_data(),
            relief="flat"
)
fin_bbutton_13.place(
            x=1177.0,
            y=813.0,
            width=30.087722778320312,
            height=35.0
)

fin_bbutton_14 = Button(fin_b,
            image=fin_bbutton_image_12,
            borderwidth=0,
            highlightthickness=0,
            command=lambda: print("button_14 clicked"),
            relief="flat"
)
fin_bbutton_14.place(
            x=1027.0,
            y=813.0,
            width=147.0,
            height=34.0
)

fin_bbutton_15 = Button(fin_b,
            image=fin_bbutton_image_7,
            borderwidth=0,
            highlightthickness=0,
            command=lambda: update_data(),
            relief="flat"
)
fin_bbutton_15.place(
            x=1703.0,
            y=813.0,
            width=30.08770751953125,
            height=35.0
)

fin_bbutton_16 = Button(fin_b,
            image=fin_bbutton_image_12,
            borderwidth=0,
            highlightthickness=0,
            command=lambda: print("button_16 clicked"),
            relief="flat"
)
```

```
fin_bbutton_16.place(
    x=1553.0,
    y=813.0,
    width=147.0,
    height=34.0
)
progress = ttk.Progressbar(fin_b, orient="horizontal", length=890, mode="determinate")
progress.place(x=467,y=1022)

### fin l

fin_l = Canvas(
    window,
    bg = "#FFFFFF",
    height = 1080,
    width = 1920,
    bd = 0,
    highlightthickness = 0,
    relief = "ridge"
)

fin_l.place(x = 30000, y = 0)
fin_limage_image_1 = PhotoImage(
    file=relative_to_assets("fin_limage_1.png"))
fin_limage_1 = fin_l.create_image(
    960.0,
    540.0,
    image=fin_limage_image_1
)

fin_lbutton_image_1 = PhotoImage(
    file=relative_to_assets("fin_lbutton_1.png"))
fin_lbutton_1 = Button(fin_l,
    image=fin_lbutton_image_1,
    borderwidth=0,
    highlightthickness=0,
    command=lambda: cur_buy(fin_l),
    relief="flat"
)
fin_lbutton_1.place(
    x=913.5242309570312,
    y=84.0,
    width=204.89328002929688,
    height=53.0
)

fin_lbutton_image_2 = PhotoImage(
    file=relative_to_assets("fin_lbutton_2.png"))
fin_lbutton_2 = Button(fin_l,
    image=fin_lbutton_image_2,
    borderwidth=0,
    highlightthickness=0,
    command=lambda: cur_slwin(fin_l),
    relief="flat"
)
fin_lbutton_2.place(
        x=1213.0,
        y=84.0,
        width=205.0,
        height=53.0
)



price_nd_namel =fin_l.create_text(
        242.0,
        536.0,
        anchor="nw",
        text="₹10,00,000",
        fill="#746C81",
        font=("Inter", 34 * -1)
)

fin_lbutton_image_3 = PhotoImage(
        file=relative_to_assets("fin_lbutton_3.png"))
fin_lbutton_3 = Button(fin_l,
        image=fin_lbutton_image_3,
        borderwidth=0,
        highlightthickness=0,
  command=lambda: changebankchosen2("State Bank of India"),
        relief="flat"
)
fin_lbutton_3.place(
        x=133.0,
        y=358.0,
        width=225.0,
        height=94.0
)

fin_lbutton_image_4 = PhotoImage(
        file=relative_to_assets("fin_lbutton_4.png"))
fin_lbutton_4 = Button(fin_l,
        image=fin_lbutton_image_4,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: changebankchosen2("South Indian Bank"),
        relief="flat"
)
fin_lbutton_4.place(
        x=381.0,
        y=358.0,
        width=225.0,
        height=94.0
)

fin_lbutton_image_5 = PhotoImage(
        file=relative_to_assets("fin_lbutton_5.png"))
fin_lbutton_5 = Button(fin_l,
        image=fin_lbutton_image_5,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: changebankchosen2("Axis Bank"),
		relief="flat"
)
fin_lbutton_5.place(
		x=629.0,
		y=358.0,
		width=225.0,
		height=94.0
)

fin_lbutton_image_6 = PhotoImage(
		file=relative_to_assets("fin_lbutton_6.png"))
fin_lbutton_6 = Button(fin_l,
		image=fin_lbutton_image_6,
		borderwidth=0,
		highlightthickness=0,
		command=lambda: changebankchosen2("HDFC Bank"),
		relief="flat"
)
fin_lbutton_6.place(
		x=877.0,
		y=358.0,
		width=225.0,
		height=94.0
)

fin_lbutton_image_7 = PhotoImage(
		file=relative_to_assets("fin_lbutton_7.png"))
fin_lbutton_7 = Button(fin_l,
		image=fin_lbutton_image_7,
		borderwidth=0,
		highlightthickness=0,
		command=lambda: print("button_7 clicked"),
		relief="flat"
)
fin_lbutton_7.place(
		x=725.0,
		y=738.0,
		width=30.087722778320312,
		height=35.0
)

fin_lbutton_image_8 = PhotoImage(
		file=relative_to_assets("fin_lbutton_8.png"))
fin_lbutton_8 = Button(fin_l,
		image=fin_lbutton_image_8,
		borderwidth=0,
		highlightthickness=0,
		command=lambda: import_files(),
		relief="flat"
)
fin_lbutton_8.place(
		x=607.0,
		y=647.0,
		width=99.0,
		height=81.0
)
```

```
fin_lbutton_9 = Button(fin_l,
        image=fin_lbutton_image_8,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: import_files(),
        relief="flat"
)
fin_lbutton_9.place(
        x=1587.0,
        y=647.0,
        width=99.0,
        height=81.0
)


fin_lbutton_10 = Button(fin_l,
        image=fin_lbutton_image_8,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: import_files(),
        relief="flat"
)
fin_lbutton_10.place(
        x=1058.0,
        y=647.0,
        width=99.0,
        height=81.0
)

fin_lentry_image_1 = PhotoImage(
        file=relative_to_assets("fin_lentry_1.png"))
fin_lentry_bg_1 = fin_l.create_image(
        809.5,
        488.0,
        image=fin_lentry_image_1
)
fin_lentry_1 = Entry(fin_l,
        bd=0,
        bg="#FFF7D7",
        fg="#000716",
        highlightthickness=0
)
fin_lentry_1.place(
        x=520.0,
        y=465.0,
        width=579.0,
        height=44.0
)

bankfirml =fin_l.create_text(
        362.0,
        308.0,
        anchor="nw",
        text="Choose a bank",
        fill="#746C81",
        font=("Inter", 32 * -1)
)

fin_lbutton_image_11 = PhotoImage(
        file=relative_to_assets("fin_lbutton_11.png"))
fin_lbutton_11 = Button(fin_l,
        image=fin_lbutton_image_11,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: cur_invoice(fin_l),
        relief="flat"
)
fin_lbutton_11.place(
        x=755.0,
        y=966.0,
        width=363.0,
        height=71.0
)

fin_lbutton_image_12 = PhotoImage(
        file=relative_to_assets("fin_lbutton_12.png"))
fin_lbutton_12 = Button(fin_l,
        image=fin_lbutton_image_12,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: print("button_12 clicked"),
        relief="flat"
)
fin_lbutton_12.place(
        x=575.0,
        y=738.0,
        width=147.0,
        height=34.0
)

fin_l.create_text(
        613.0,
        830.0,
        anchor="nw",
        text="Upload Bank Transactions: ",
        fill="#746C81",
        font=("Inter Medium", 34 * -1)
)


fin_lbutton_13 = Button(fin_l,
        image=fin_lbutton_image_7,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: print("button_13 clicked"),
        relief="flat"
)
fin_lbutton_13.place(
        x=1177.0,
        y=905.0,
        width=30.08770751953125,
        height=35.0
)


fin_lbutton_14 = Button(fin_l,
        image=fin_lbutton_image_8,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: import_files(),
        relief="flat"
)
fin_lbutton_14.place(
        x=1059.0,
        y=814.0,
        width=99.0,
        height=81.0
)

fin_lbutton_15 = Button(fin_l,
        image=fin_lbutton_image_12,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: print("button_15 clicked"),
        relief="flat"
)
fin_lbutton_15.place(
        x=1027.0,
        y=905.0,
        width=147.0,
        height=34.0
)

fin_lbutton_16 = Button(fin_l,
        image=fin_lbutton_image_7,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: print("button_16 clicked"),
        relief="flat"
)
fin_lbutton_16.place(
        x=1177.0,
        y=738.0,
        width=30.08770751953125,
        height=35.0
)

fin_lbutton_17 = Button(fin_l,
        image=fin_lbutton_image_12,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: print("button_17 clicked"),
        relief="flat"
)
fin_lbutton_17.place(
        x=1027.0,
        y=738.0,
        width=147.0,
        height=34.0
)

fin_lbutton_18 = Button(fin_l,
        image=fin_lbutton_image_7,
        borderwidth=0,
   highlightthickness=0,
        command=lambda: print("button_18 clicked"),
        relief="flat"
)
fin_lbutton_18.place(
        x=1703.0,
        y=738.0,
        width=30.08770751953125,
        height=35.0
)

fin_lbutton_19 = Button(fin_l,
        image=fin_lbutton_image_12,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: print("button_19 clicked"),
        relief="flat"
)
fin_lbutton_19.place(
        x=1553.0,
        y=738.0,
        width=147.0,
        height=34.0
)

fin_lprogress = ttk.Progressbar(fin_l, orient="horizontal", length=200, mode="determinate")





sel_1 = Canvas(
        window,
        bg = "#FFFFFF",
        height = 1080,
        width = 1920,
        bd = 0,
        highlightthickness = 0,
        relief = "ridge"
)

sel_1.place(x = 3000, y = 3000)
sel_1image_image_1 = PhotoImage(
        file=relative_to_assets("sel_1image_1.png"))
sel_1image_1 = sel_1.create_image(
        960.0,
        540.0,
        image=sel_1image_image_1
)

sel_1button_image_1 = PhotoImage(
        file=relative_to_assets("sel_1button_1.png"))
sel_1button_1 = Button(sel_1,
        image=sel_1button_image_1,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: cur_buy(sel_1),
        relief="flat"
)
sel_1button_1.place(
        x=913.524169921875,
        y=84.0,
        width=204.893310546875,
        height=53.0
)

sel_1button_image_2 = PhotoImage(
        file=relative_to_assets("sel_1button_2.png"))
sel_1button_2 = Button(sel_1,
        image=sel_1button_image_2,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: cur_slwin(sel_1),
        relief="flat"
)
sel_1button_2.place(
        x=1213.0,
  y=84.0,
        width=205.0,
        height=53.0
)

sel_1button_image_3 = PhotoImage(
        file=relative_to_assets("sel_1button_3.png"))
sel_1button_3 = Button(sel_1,
        image=sel_1button_image_3,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: sel_1_sel_2(),
        relief="flat"
)
sel_1button_3.place(
        x=792,
        y=964,
        width=363,
        height=71.0
)

sl_1historycombo = ttk.Combobox(sel_1,width=28, value =car_attributes.get('History') )
sl_1historycombo.place(x=178.0, y=328.0)

sl_1kmdrivencombo = ttk.Combobox(sel_1,width=28, value =car_attributes.get('kmsdriven') )
sl_1kmdrivencombo.place(x=521.0, y=328.0)
```

```
sl_1last_servicecombo = ttk.Combobox(sel_1,width=28, value =car_attributes.get('lastservice') )
sl_1last_servicecombo.place(x=864.0, y=328.0)

sl_1registrationcombo = ttk.Combobox(sel_1,width=28, value =car_attributes.get('registered_in') )
sl_1registrationcombo.place(x=1207.0, y=328.0)

sl_1registeredincombo = ttk.Entry(sel_1,width=28,)
sl_1registeredincombo.place(x=1550.0, y=328.0)

sl_1fueltypecombo = ttk.Combobox(sel_1,width=28, value =car_attributes.get('fuel_type') )
sl_1fueltypecombo.place(x=178.0, y=581.0)

sl_1enginetypecombo = ttk.Combobox(sel_1,width=28, value =car_attributes.get('engine_type') )
sl_1enginetypecombo.place(x=521.0, y=581.0)

sl_1transmissiontypecombo = ttk.Combobox(sel_1,width=28, value =car_attributes.get('transmission') )
sl_1transmissiontypecombo.place(x=864.0, y=581.0)

sl_1drivetraincombo = ttk.Combobox(sel_1,width=28, value =car_attributes.get('drivetrain') )
sl_1drivetraincombo.place(x=1207.0, y=581.0)

sl_1ownercombo = ttk.Combobox(sel_1,width=28, value =car_attributes.get('owner') )
sl_1ownercombo.place(x=1550.0, y=581.0)

sl_1insurancecombo = ttk.Combobox(sel_1,width=28, value =car_attributes.get('insurance') )
sl_1insurancecombo.place(x=178.0, y=834.0)

sl_1airbagscombo = ttk.Combobox(sel_1,width=28, value =car_attributes.get('airbags') )
sl_1airbagscombo.place(x=521.0, y=834.0)

sl_1mileagecombo = ttk.Combobox(sel_1,width=28, value =car_attributes.get('mileage') )
sl_1mileagecombo.place(x=864.0, y=834.0)

sl_1steeringcombo = ttk.Combobox(sel_1,width=28, value =car_attributes.get('steering_type') )
sl_1steeringcombo.place(x=1207.0, y=834.0)

sl_1seatupholstrycombo = ttk.Combobox(sel_1,width=28, value =car_attributes.get('seat_upholstery') )
sl_1seatupholstrycombo.place(x=1550.0, y=834.0)


sel_2 = Canvas(
        window,
        bg = "#FFFFFF",
        height = 1080,
        width = 1920,
        bd = 0,
        highlightthickness = 0,
        relief = "ridge"
)

sel_2.place(x = 3000, y = 0)
sel_2image_image_1 = PhotoImage(
        file=relative_to_assets("sel_2image_1.png"))
sel_2image_1 = sel_2.create_image(
        960.0,
        540.0,
        image=sel_2image_image_1
)

sel_2button_image_1 = PhotoImage(
        file=relative_to_assets("sel_2button_1.png"))
sel_2button_1 = Button(sel_2,
        image=sel_2button_image_1,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: print("sel_2button_1 clicked"),
        relief="flat"
)
sel_2button_1.place(
        x=1213.0,
        y=84.0,
        width=205.0,
        height=53.0
)

sel_2button_image_2 = PhotoImage(
        file=relative_to_assets("sel_2button_2.png"))
sel_2button_2 = Button(sel_2,
        image=sel_2button_image_2,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: print("sel_2button_2 clicked"),
        relief="flat"
)
#sel_2button_2.place(
#       x=1414.0,
#       y=951.0,
#       width=30.0877685546875,
#       height=35.0
#)

sel_2button_image_3 = PhotoImage(
        file=relative_to_assets("sel_2button_3.png"))
sel_2button_3 = Button(sel_2,
        image=sel_2button_image_3,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: uploadpic(sel_2button_3),
        relief="flat"
)
sel_2button_3.place(
        x=1296.0,
        y=860.0,
        width=99.0,
        height=81.0
)

sel_2button_image_4 = PhotoImage(
        file=relative_to_assets("sel_2button_4.png"))
sel_2button_4 = Button(sel_2,
        image=sel_2button_image_4,
            borderwidth=0,
            highlightthickness=0,
            command=lambda: print("sel_2button_4 clicked"),
            relief="flat"
)
#sel_2button_4.place(
#           x=1264.0,
#           y=951.0,
#           width=147.0,
#           height=34.0
#)


sel_2button_5 = Button(sel_2,
            image=sel_2button_image_2,
            borderwidth=0,
            highlightthickness=0,
            command=lambda: print("sel_2button_5 clicked"),
            relief="flat"
)
#sel_2button_5.place(
#           x=1747.0,
#    y=951.0,
#           width=30.0877685546875,
#           height=35.0
#)


sel_2button_6 = Button(sel_2,
            image=sel_2button_image_3,
            borderwidth=0,
            highlightthickness=0,
            command=lambda: uploadpic(sel_2button_6),
            relief="flat"
)
sel_2button_6.place(
            x=1629.0,
            y=860.0,
            width=99.0,
            height=81.0
)

sel_2button_7 = Button(sel_2,
            image=sel_2button_image_4,
            borderwidth=0,
            highlightthickness=0,
            command=lambda: print("sel_2button_7 clicked"),
            relief="flat"
)

orvmscombo = ttk.Combobox(sel_2,width=28, value = car_attributes.get('ORVMs'))
orvmscombo.place(x=178.0, y=328.0)

enginestart_topcombo = ttk.Combobox(sel_2,width=28, value = car_attributes.get('engine_start_stop'))
enginestart_topcombo.place(x=521.0, y=328.0)
```

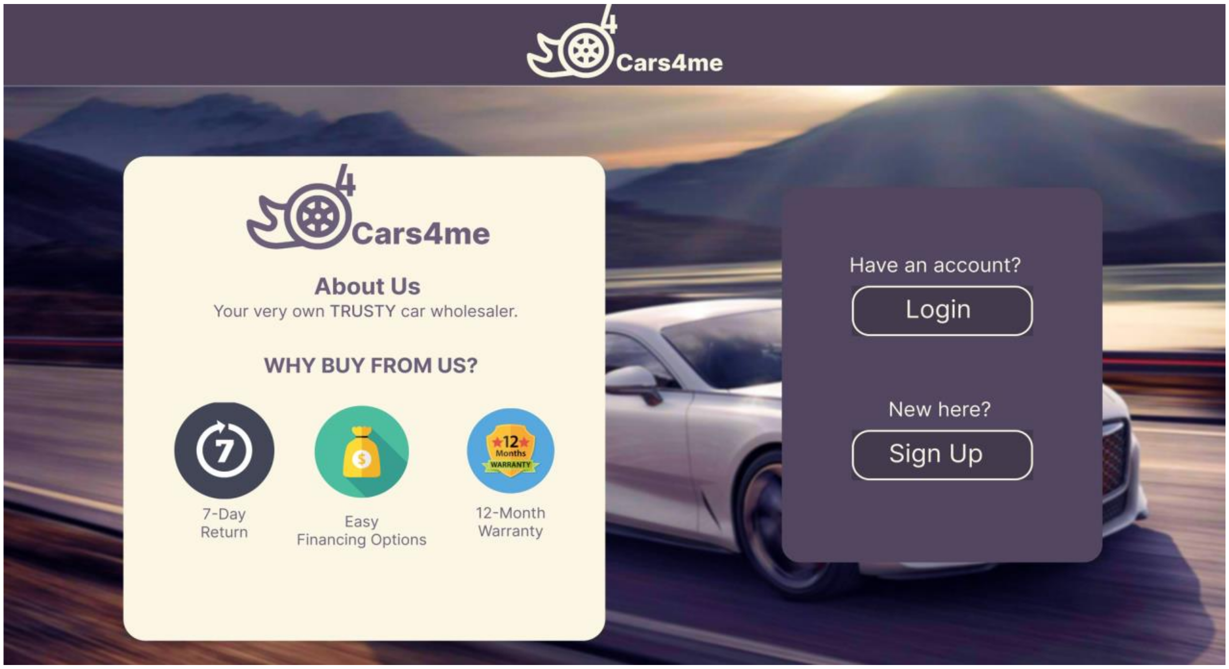
```
centrallockingcombo = ttk.Combobox(sel_2,width=28, value = car_attributes.get('central_locking'))
centrallockingcombo.place(x=864.0, y=328.0)

sunroofcombo = ttk.Combobox(sel_2,width=28, value = car_attributes.get('sunroof'))
sunroofcombo.place(x=1207.0, y=328.0)

rearaccombo = ttk.Combobox(sel_2,width=28, value = car_attributes.get('rear_ac'))
rearaccombo.place(x=1550.0, y=328.0)

powerwindowscombo = ttk.Combobox(sel_2,width=28, value = car_attributes.get('power_windows'))
powerwindowscombo.place(x=178.0, y=581.0)

headlampscombo = ttk.Combobox(sel_2,width=28, value = car_attributes.get('headlamps'))
headlampscombo.place(x=521.0, y=581.0)

maxpowercombo = ttk.Combobox(sel_2,width=28, value = car_attributes.get('max_power'))
maxpowercombo.place(x=864.0, y=581.0)

fueltankcapcombo = ttk.Combobox(sel_2,width=28, value = car_attributes.get('Fuel_capacity'))
fueltankcapcombo.place(x=1207.0, y=581.0)

seatingcapacitycombo = ttk.Combobox(sel_2,width=28, value = car_attributes.get('seating_capacity'))
seatingcapacitycombo.place(x=1550.0, y=581.0)

alterantefueltypecombo = ttk.Combobox(sel_2,width=28, value = car_attributes.get('alternate_fuel') )
alterantefueltypecombo.place(x=178.0, y=834.0)

musicsystemcombo = ttk.Combobox(sel_2,width=28, value = car_attributes.get('music_system') )
musicsystemcombo.place(x=521.0, y=834.0)

transmissioncombo = ttk.Combobox(sel_2,width=28, value = car_attributes.get('transmission') )
transmissioncombo.place(x=864.0, y=834.0)



sel_2button_image_8 = PhotoImage(
        file=relative_to_assets("sel_2button_8.png"))
sel_2button_8 = Button(sel_2,
        image=sel_2button_image_8,
        borderwidth=0,
        highlightthickness=0,
        command=lambda: sel_2_toselling(),
        relief="flat"
)
sel_2button_8.place(
        x=792,
        y=964,
        width=363,
        height=71.0
)


invoice = Canvas(
        window,
        bg = "#FFFFFF",
        height = 1080,
        width = 1920,
        bd = 0,
        highlightthickness = 0,
        relief = "ridge"
)

invoice.place(x = 3000, y = 3000)
invoiceimage_image_1 = PhotoImage(
        file=relative_to_assets("invoiceimage_1.png"))
invoiceimage_1 = invoice.create_image(
        961.0,
        540.0,
        image=invoiceimage_image_1
)

iname =invoice.create_text(
        798.0,
        286.0,
        anchor="nw",
        text="ABCD123",
        fill="#FFF7D7",
        font=("Inter Regular", 33 * -1)
)

idate =invoice.create_text(
        1585.0,
        286.0,
        anchor="nw",
        text=(datetime.now()).strftime("%B %d, %Y"),
        fill="#FFF7D7",
        font=("Inter Regular", 33 * -1)
)

imobileno =invoice.create_text(
        1235.0,
        286.0,
        anchor="nw",
  text="2131322311",
        fill="#FFF7D7",
        font=("Inter Regular", 33 * -1)
)

iregistration = invoice.create_text(
        798.0,
        451.0,
        anchor="nw",
        text="TN54M4315",
        fill="#FFF7D7",
        font=("Inter Regular", 33 * -1)
)

imodelyr = invoice.create_text(
        1669.0,
        451.0,
        anchor="nw",
        text="2017",
        fill="#FFF7D7",
        font=("Inter Regular", 33 * -1)
)

imodel = invoice.create_text(
        1121.0,
        556.0,
        anchor="nw",
        text="Civic",
        fill="#FFF7D7",
        font=("Inter Regular", 33 * -1)
)

imileage =invoice.create_text(
        1423.0,
        556.0,
        anchor="nw",
        text="15 - 20 km/l",
        fill="#FFF7D7",
        font=("Inter Regular", 33 * -1)
)

imake = invoice.create_text(
        1274.0,
        451.0,
        anchor="nw",
        text="Honda",
        fill="#FFF7D7",
        font=("Inter Regular", 33 * -1)
)

imethod =invoice.create_text(
        798.0,
        720.0,
        anchor="nw",
        text="Bank",
        fill="#FFF7D7",
        font=("Inter Regular", 33 * -1)
)

itax= invoice.create_text(
        1649.0,
        720.0,
        anchor="nw",
        text="15%",
        fill="#FFF7D7",
        font=("Inter Regular", 33 * -1)
)

iprice =invoice.create_text(
        1220.0,
        720.0,
        anchor="nw",
        text="4,00,000",
        fill="#FFF7D7",
        font=("Inter Regular", 33 * -1)
```

```
)

itotal =invoice.create_text(
            1226.0,
            818.0,
            anchor="nw",
            text="4,21,000",
            fill="#FFF7D7",
            font=("Inter Regular", 33 * -1)
)

invoicebutton_image_1 = PhotoImage(
            file=relative_to_assets("invoicebutton_1.png"))
invoicebutton_1 = Button(invoice,
            image=invoicebutton_image_1,
            borderwidth=0,
            highlightthickness=0,
            command=lambda: cur_buy(invoice),
            relief="flat"
)
invoicebutton_1.place(
            x=989.0,
            y=964.0,
            width=274.0,
            height=65.0
)

invoicebutton_image_2 = PhotoImage(
            file=relative_to_assets("invoicebutton_2.png"))
invoicebutton_2 = Button(invoice,
            image=invoicebutton_image_2,
            borderwidth=0,
            highlightthickness=0,
            command=lambda: invoicedownload(),
            relief="flat"
)
invoicebutton_2.place(
            x=1371.0,
            y=964.0,
            width=274.0,
            height=65.0
)
window.resizable(True,True)
window.mainloop()

mydb.close()
```

OUTPUT

OUTPUT

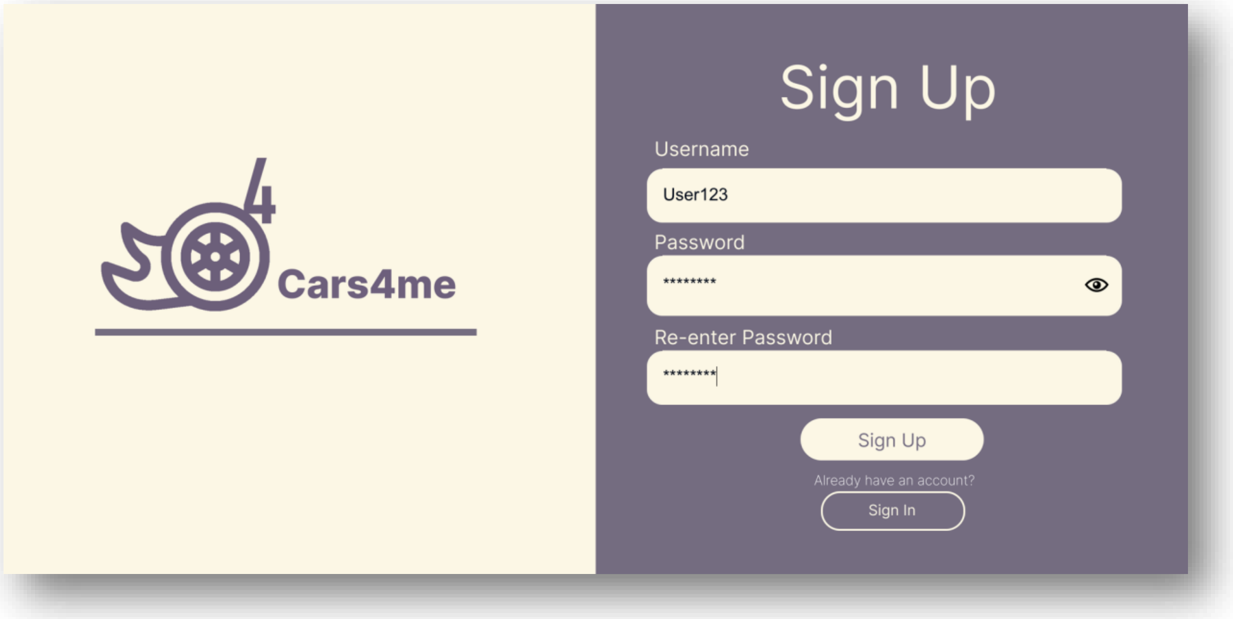
OPENING WINDOW:



Your very own TRUSTY car wholesaler.

This is the screen that appears when you initially open the application.

SIGN UP WINDOW:



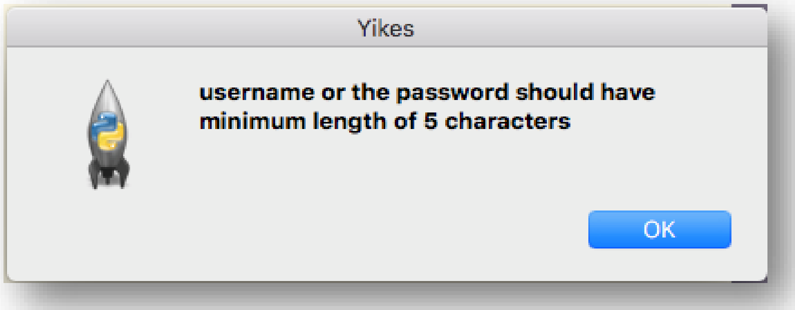
Username: User123

Password: ********

Re-enter Password: ********

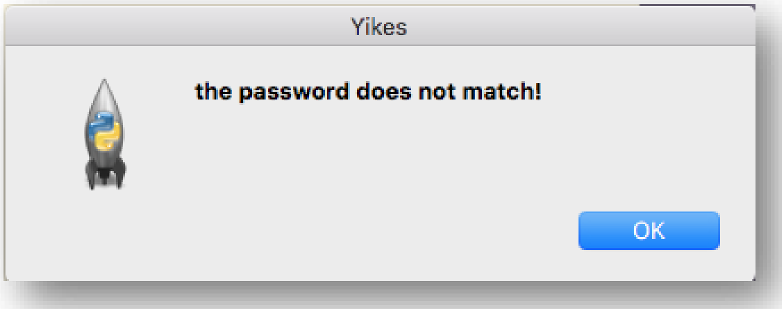
The sign up page is shown above. The user can create the account.

SIGN UP WINDOW: WRONG CREDENTIALS



Yikes

username or the password should have minimum length of 5 characters

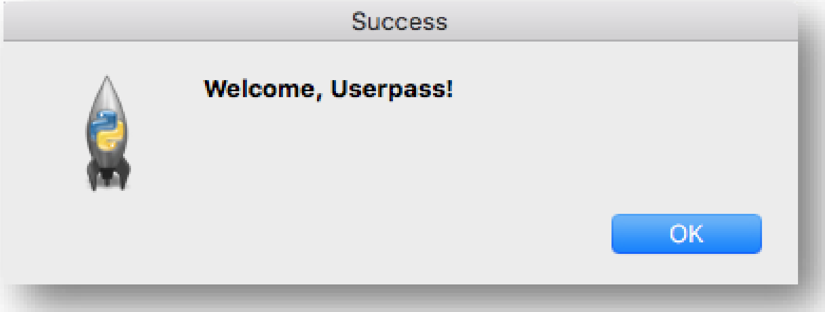


Yikes

the password does not match!

When the wrong credentials are entered, the above message boxes are shown.

SIGN UP WINDOW: CORRECT CREDENTIALS

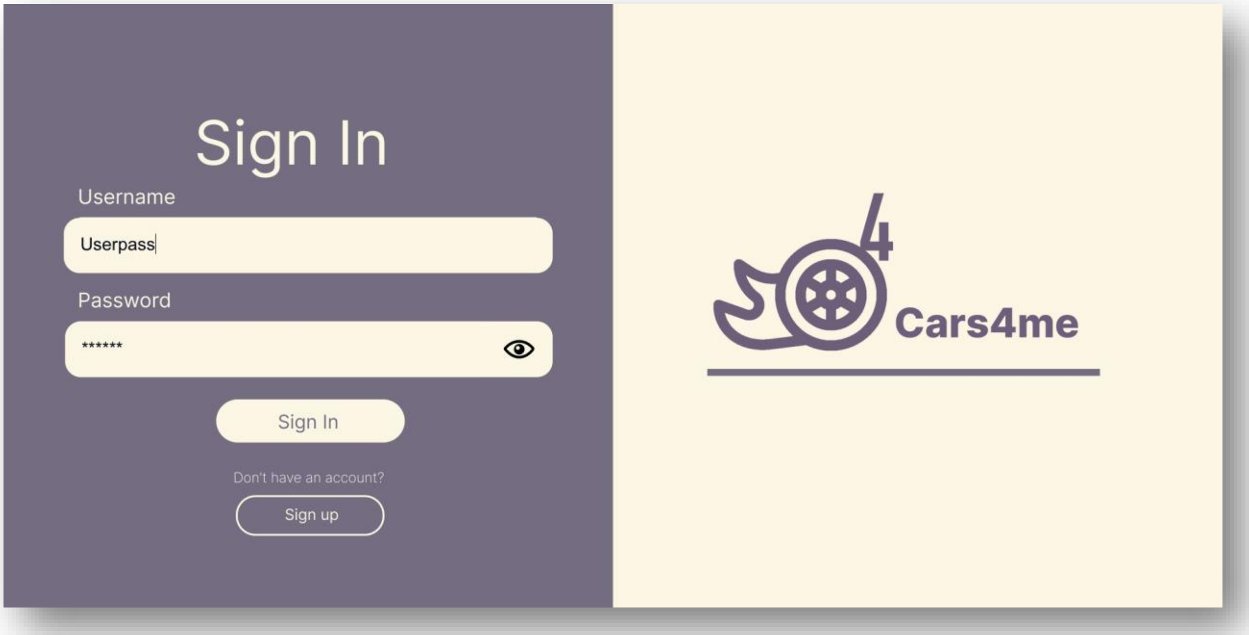


Success

Welcome, Userpass!

When the right credentials are entered, the above message boxes are shown.

SIGN IN WINDOW:



Username

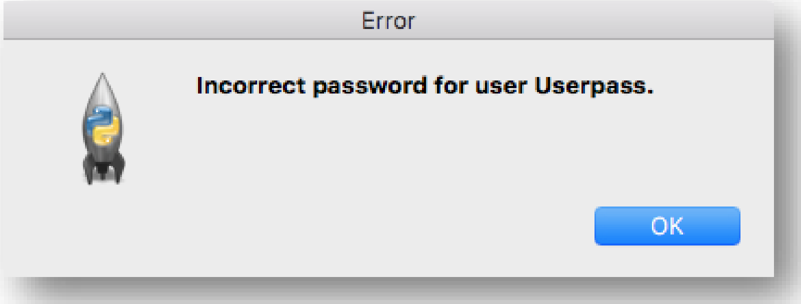
Userpass

Password

******

The sign in page is shown below. The user can login to their account.

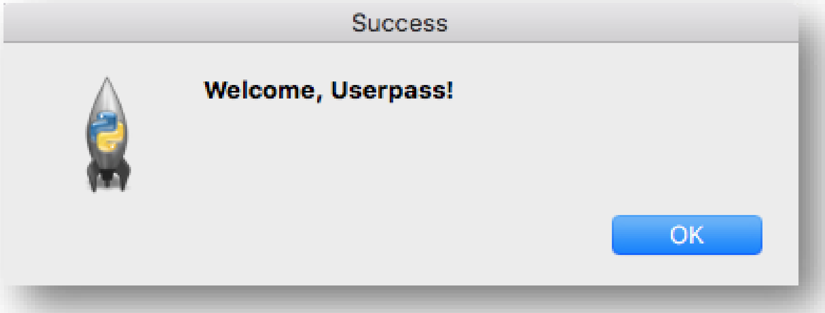
SIGN IN WINDOW: WRONG CREDENTIALS



Incorrect password for user Userpass.

When the wrong credentials are entered, the above message boxes are shown.

SIGN IN WINDOW: CORRECT CREDENTIALS

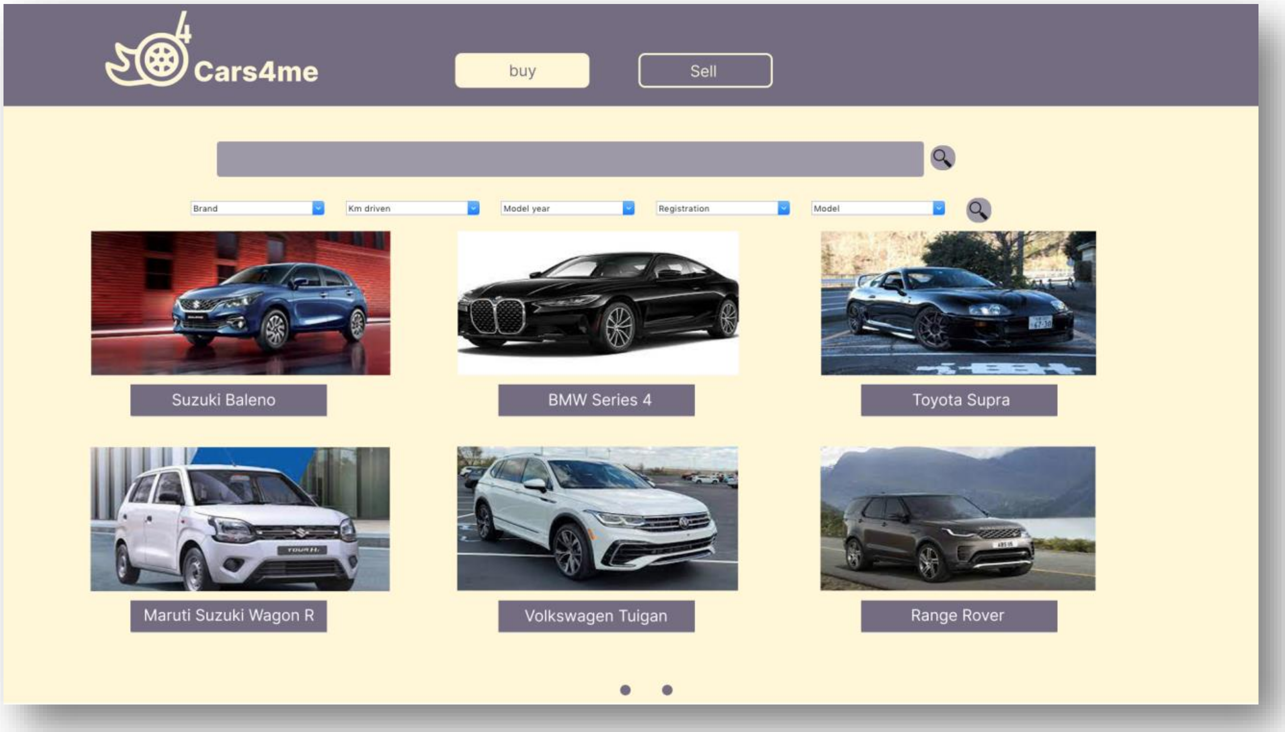


Success

Welcome, Userpass!

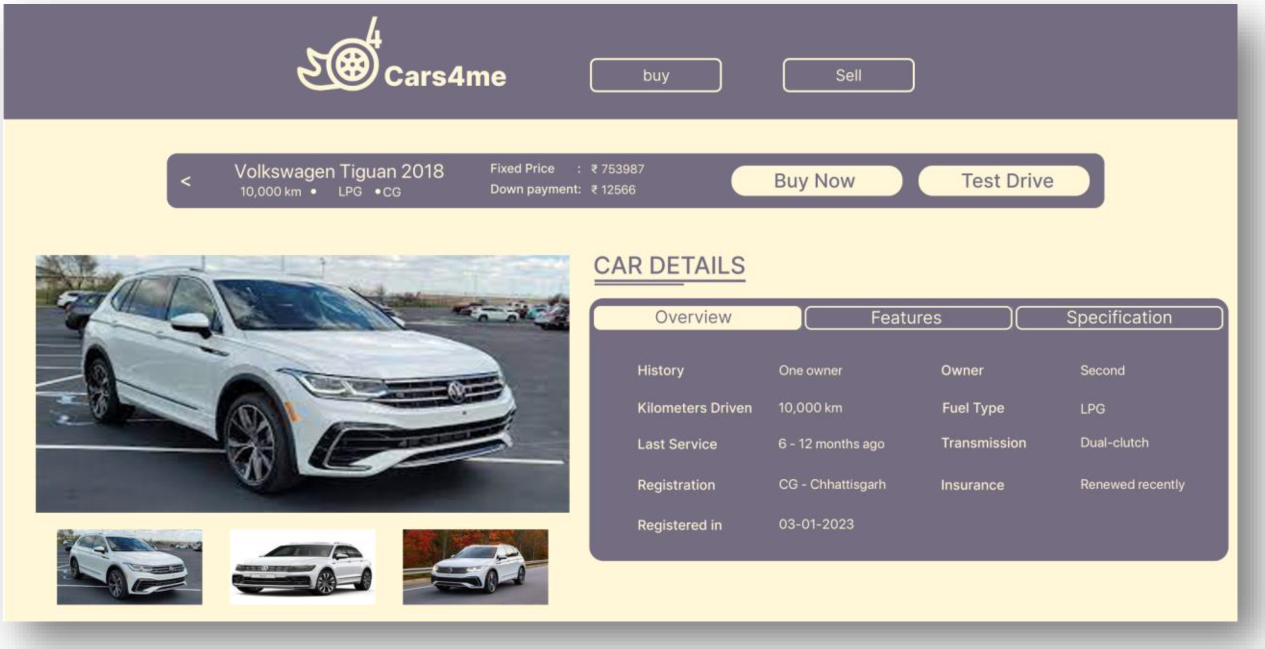
When the right credentials are entered, the above message boxes are shown.

MAIN WINDOW:



The above is the main page using which the user can access all the aims of the application. This page primarily for buying the listed cars. User can click the car to go into viewing and information page of the car. Or user can click sell button on the topbar to navigate to selling page of the car.

VIEWING AND INFORMATION PAGE:



Volkswagen Tiguan 2018
10,000 km • LPG •CG

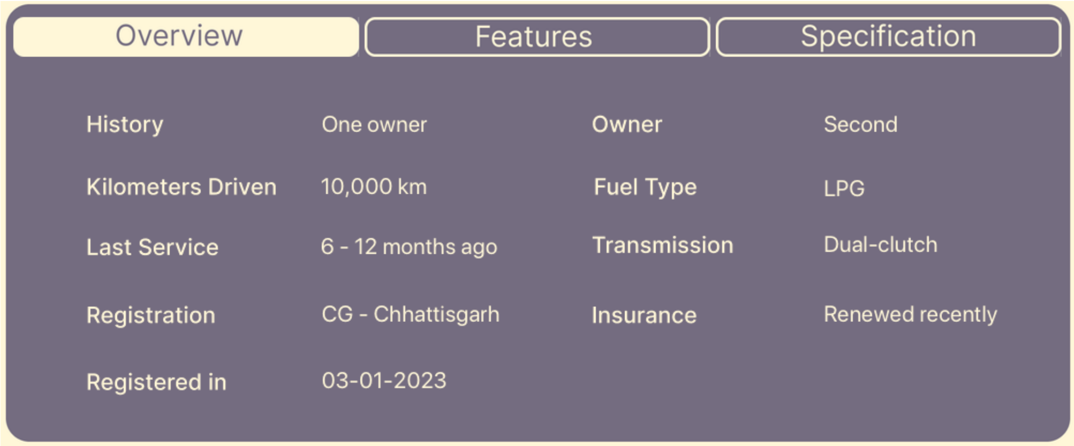
Fixed Price : ₹ 753987
Down payment: ₹ 12566

CAR DETAILS

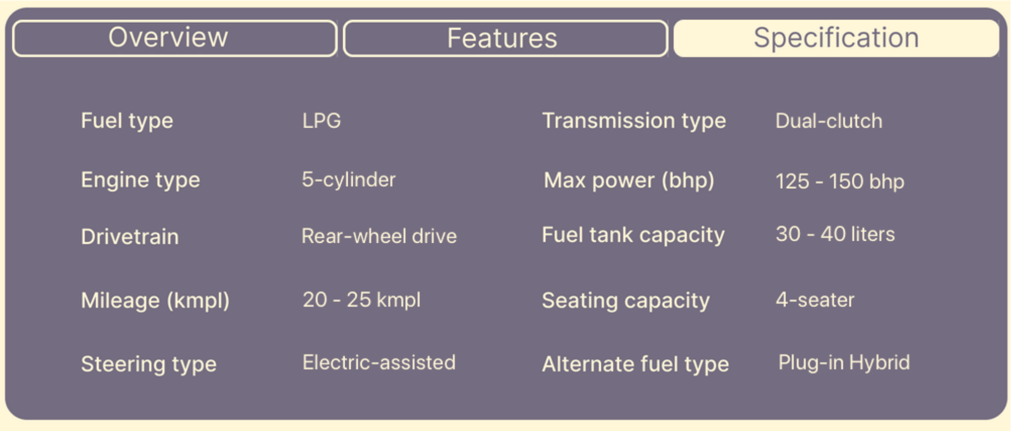
Overview	Features	Specification	
History	One owner	Owner	Second
Kilometers Driven	10,000 km	Fuel Type	LPG
Last Service	6 - 12 months ago	Transmission	Dual-clutch
Registration	CG - Chhattisgarh	Insurance	Renewed recently
Registered in	03-01-2023		

This page has 3 images of the car, which can be viewed when clicked on it. Great vast information of the car is displayed on the page.

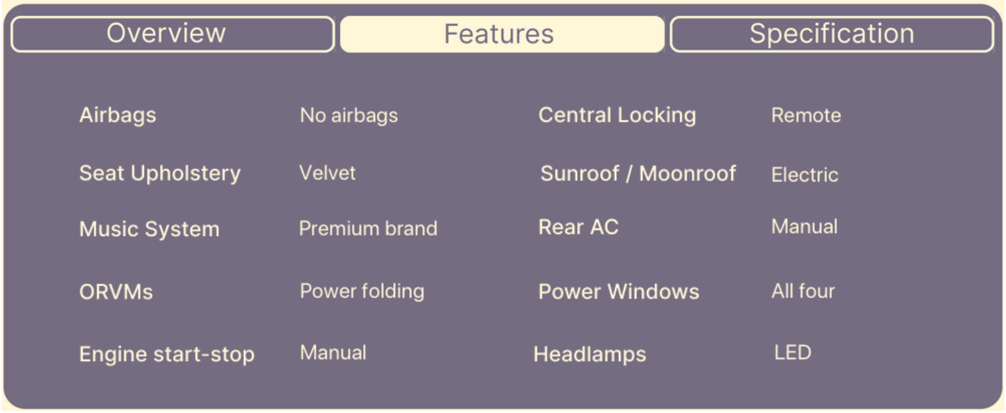
DIFFERENT DATA ABOUT THE CAR:



Overview	Features	Specification	
History	One owner	Owner	Second
Kilometers Driven	10,000 km	Fuel Type	LPG
Last Service	6 - 12 months ago	Transmission	Dual-clutch
Registration	CG - Chhattisgarh	Insurance	Renewed recently
Registered in	03-01-2023		

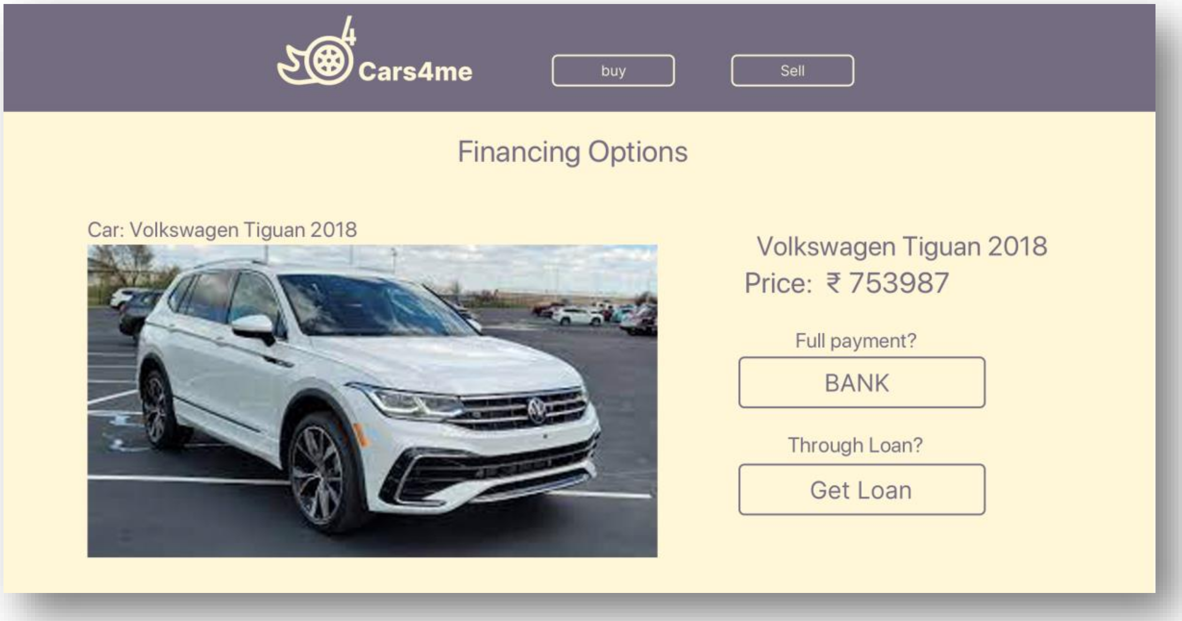


Fuel type	LPG	Transmission type	Dual-clutch
Engine type	5-cylinder	Max power (bhp)	125 - 150 bhp
Drivetrain	Rear-wheel drive	Fuel tank capacity	30 - 40 liters
Mileage (kmpl)	20 - 25 kmpl	Seating capacity	4-seater
Steering type	Electric-assisted	Alternate fuel type	Plug-in Hybrid



Airbags	No airbags	Central Locking	Remote
Seat Upholstery	Velvet	Sunroof / Moonroof	Electric
Music System	Premium brand	Rear AC	Manual
ORVMs	Power folding	Power Windows	All four
Engine start-stop	Manual	Headlamps	LED

FINANCING OPTIONS WINDOW:



Financing Options

Car: Volkswagen Tiguan 2018

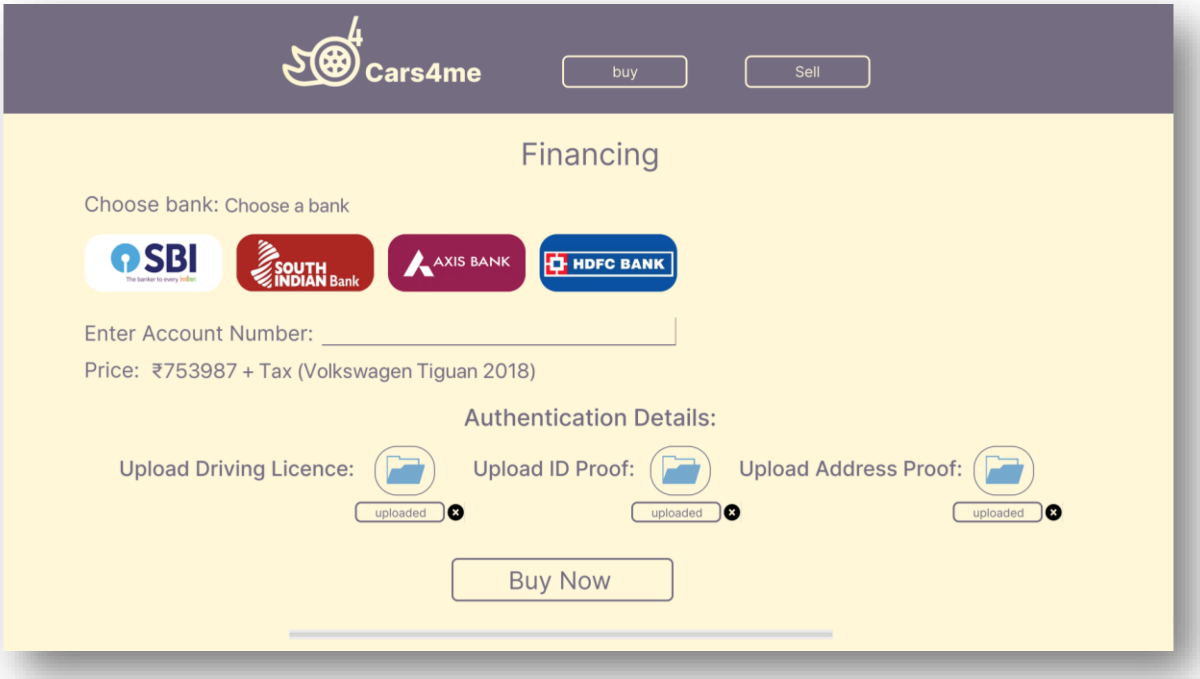
Volkswagen Tiguan 2018
Price: ₹ 753987

Full payment?

Through Loan?

This gives you two methods to buy the car.

FINANCING OPTIONS (BANK):



Financing

Choose bank: Choose a bank

Enter Account Number:

Price: ₹753987 + Tax (Volkswagen Tiguan 2018)

Authentication Details:

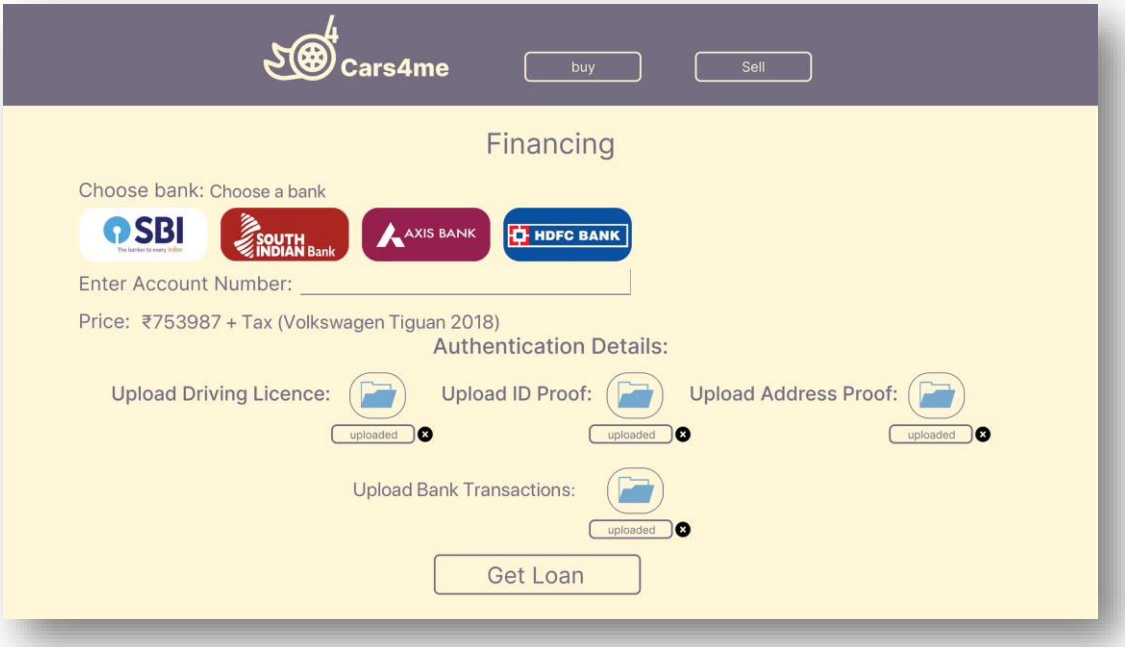
Upload Driving Licence:

Upload ID Proof:

Upload Address Proof:

The above page appear when “BANK” button is clicked.

FINANCING OPTIONS (LOAN):



Financing

Choose bank: Choose a bank

Enter Account Number:

Price: ₹753987 + Tax (Volkswagen Tiguan 2018)

Authentication Details:

Upload Driving Licence:

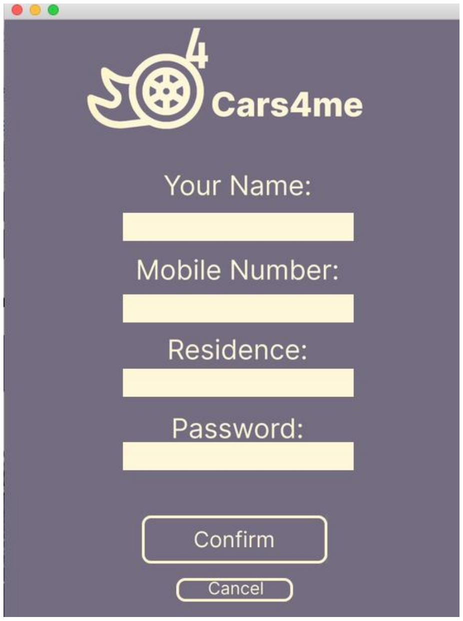
Upload ID Proof:

Upload Address Proof:

Upload Bank Transactions:

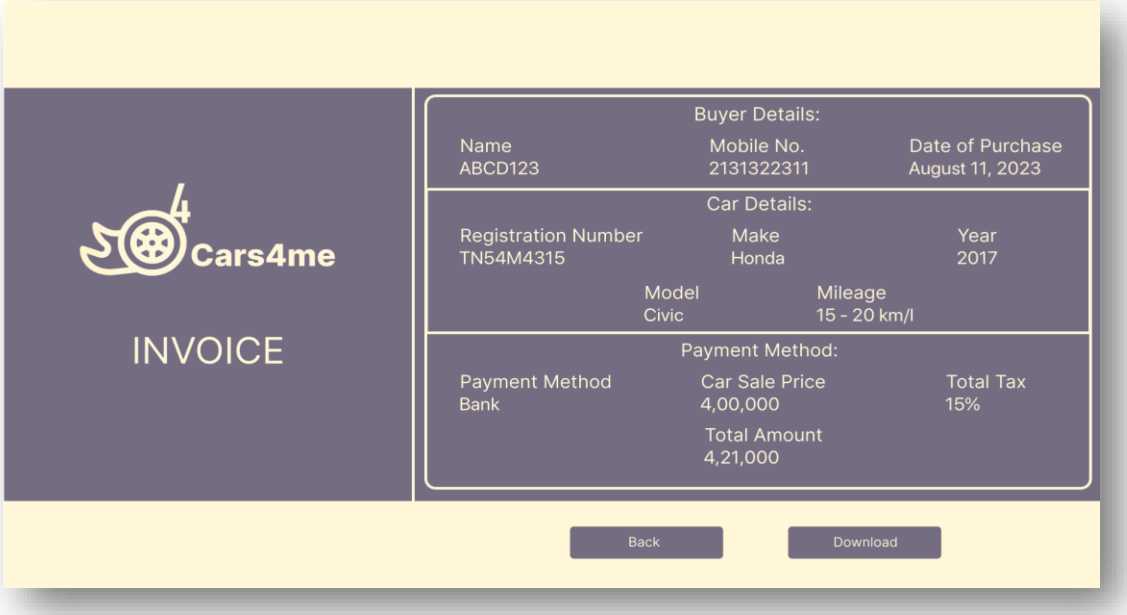
The above page appear when “BANK” button is clicked.

CONFIRMATION WINDOW:



Confirmation window appear when you have entered the needed information in bank page or the get loan page.

INVOICE WINDOW:



Buyer Details:

Name
ABCD123

Mobile No.
2131322311

Date of Purchase
August 11, 2023

Car Details:

Registration Number
TN54M4315

Make
Honda

Year
2017

Model
Civic

Mileage
15 - 20 km/l

Payment Method:

Payment Method
Bank

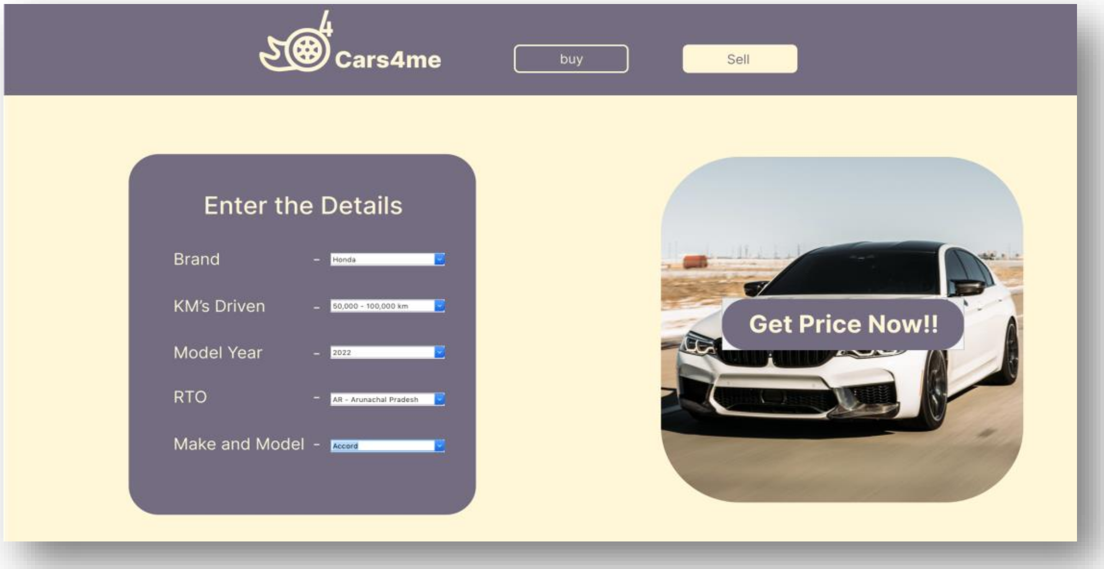
Car Sale Price
4,00,000

Total Tax
15%

Total Amount
4,21,000

After all the process you get to this invoice page. You can get the invoice downloaded into your file by clicking “Download” button.

ADD CAR INFO WINDOW: ADDING CAR:



Enter the Details

Brand - Honda

KM's Driven - 50,000 - 100,000 km

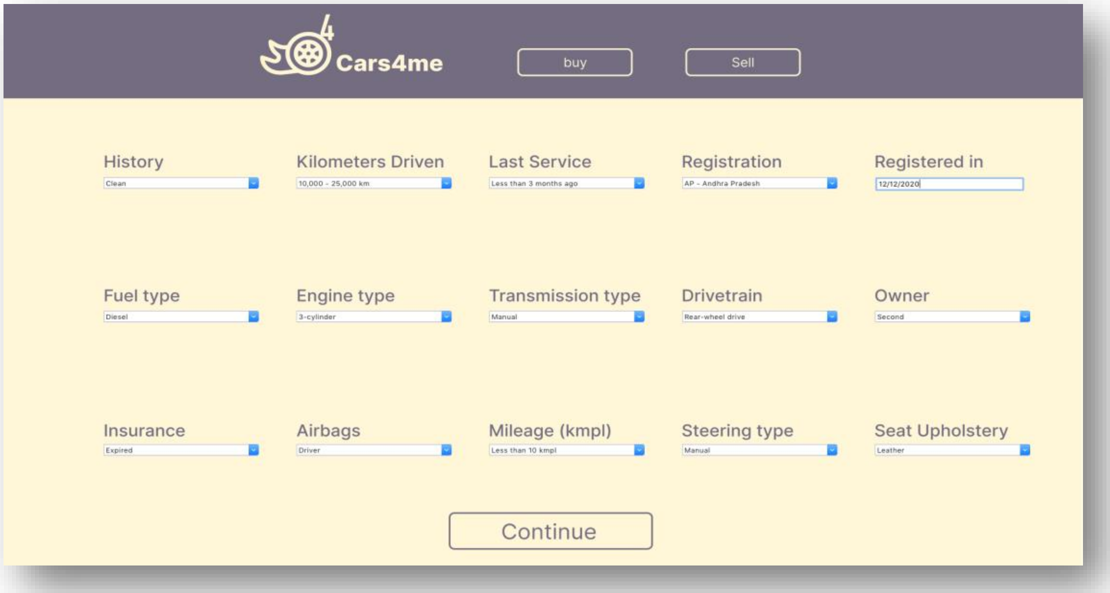
Model Year - 2022

RTO - AR - Arunachal Pradesh

Make and Model - Accord

This page appears when you click the “Sell” button on the topbar.

INFORMATION (1/2) WINDOW:



History: Clean

Kilometers Driven: 10,000 - 25,000 km

Last Service: Less than 3 months ago

Registration: AP - Andhra Pradesh

Registered in: 12/12/2020

Fuel type: Diesel

Engine type: 3-cylinder

Transmission type: Manual

Drivetrain: Rear-wheel drive

Owner: Second

Insurance: Expired

Airbags: Driver

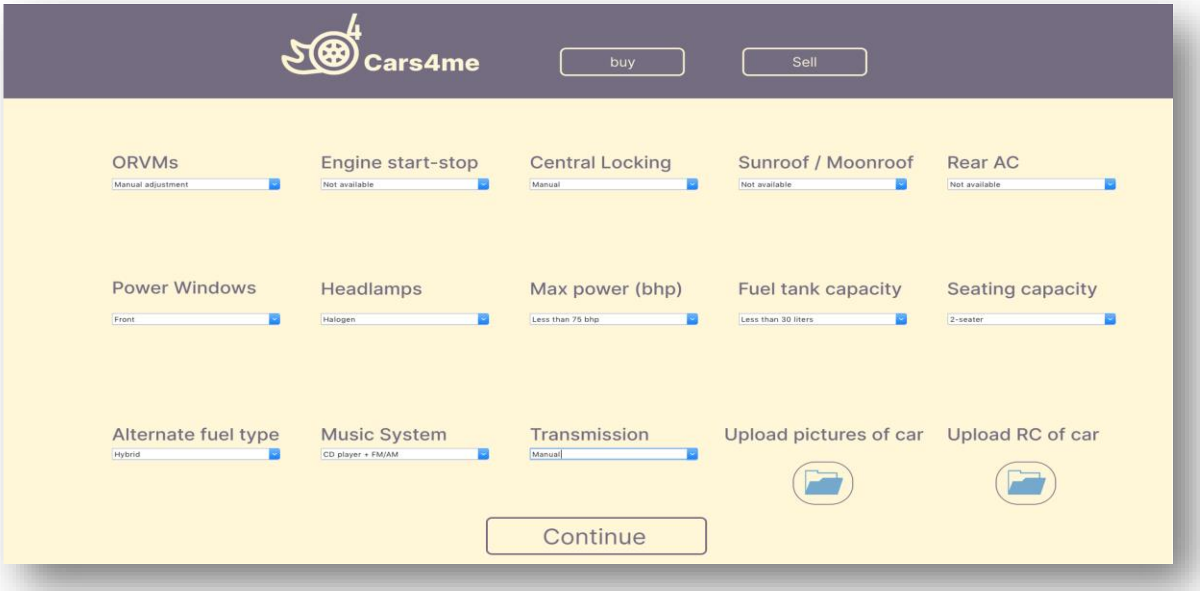
Mileage (kmpl): Less than 10 kmpl

Steering type: Manual

Seat Upholstery: Leather

This page appears when you click the “Get Price Now” button. It ask for various information about the car

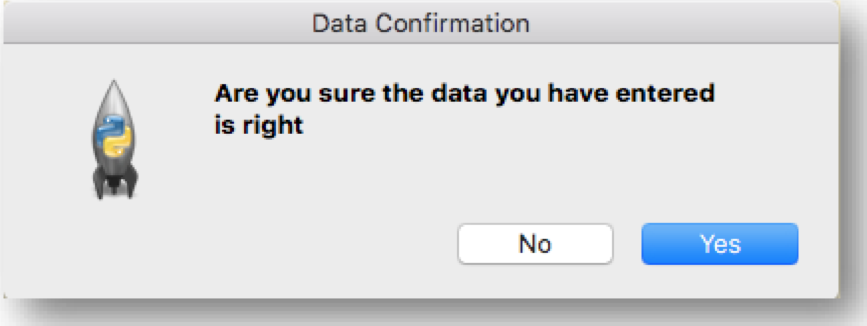
INFORMATION (2/2) WINDOW:



ORVMs: Manual adjustment
Engine start-stop: Not available
Central Locking: Manual
Sunroof / Moonroof: Not available
Rear AC: Not available
Power Windows: Front
Headlamps: Halogen
Max power (bhp): Less than 75 bhp
Fuel tank capacity: Less than 30 liters
Seating capacity: 2-seater
Alternate fuel type: Hybrid
Music System: CD player + FM/AM
Transmission: Manual
Upload pictures of car
Upload RC of car

This page appears when you click the “Continue” button. It ask for various information about the car and picture of the car

CONFIRMATION BOX (1/2):

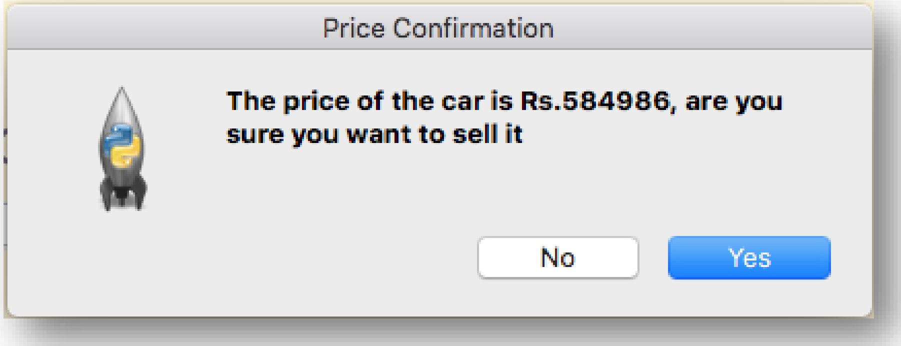


Data Confirmation

Are you sure the data you have entered is right

Asks for confirmation of the data entered.

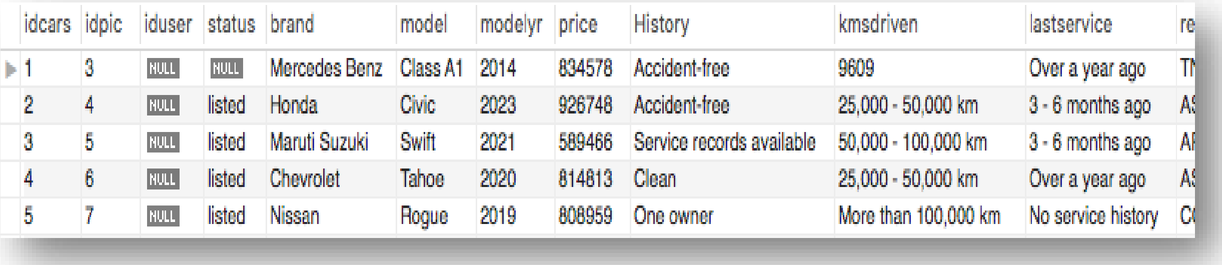
CONFIRMATION BOX (2/2):



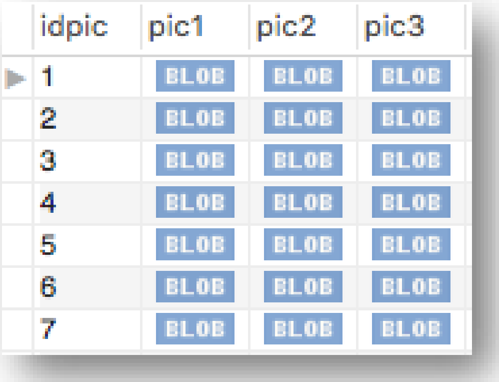
Price Confirmation

The price of the car is Rs.584986, are you sure you want to sell it

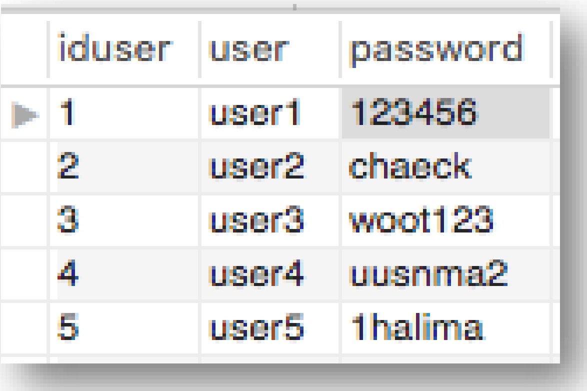
Asks for confirmation of the car price. If yes then it gets added to the mysql table



idcars	idpic	iduser	status	brand	model	modelyr	price	History	kmsdriven	lastservice	re
1	3	NULL	NULL	Mercedes Benz	Class A1	2014	834578	Accident-free	9609	Over a year ago	TI
2	4	NULL	listed	Honda	Civic	2023	926748	Accident-free	25,000 - 50,000 km	3 - 6 months ago	AS
3	5	NULL	listed	Maruti Suzuki	Swift	2021	589466	Service records available	50,000 - 100,000 km	3 - 6 months ago	AF
4	6	NULL	listed	Chevrolet	Tahoe	2020	814813	Clean	25,000 - 50,000 km	Over a year ago	AS
5	7	NULL	listed	Nissan	Rogue	2019	808959	One owner	More than 100,000 km	No service history	C



idpic	pic1	pic2	pic3
1	BLOB	BLOB	BLOB
2	BLOB	BLOB	BLOB
3	BLOB	BLOB	BLOB
4	BLOB	BLOB	BLOB
5	BLOB	BLOB	BLOB
6	BLOB	BLOB	BLOB
7	BLOB	BLOB	BLOB



iduser	user	password
1	user1	123456
2	user2	chaeck
3	user3	woot123
4	user4	uusnma2
5	user5	1halima

CONCLUSION

CONCLUSION

In conclusion, the culmination of this car marketplace application project represents a significant stride forward in the realm of modern digital solutions. The application has been meticulously crafted to cater to the specific needs of car buyers and sellers, providing a seamless platform for their transactions.

Throughout this project, we have traversed the path from concept to execution, gaining invaluable insights into the complexities of software development along the way.

As we reflect on this journey, it becomes evident that collaboration, creative problem-solving, and adaptability are key components of successful software development. The project emphasizes the significance of data security and user experience, both of which are integral to establishing trust within a digital marketplace.

Looking ahead, the application holds the potential to significantly influence how car transactions are conducted in the digital age. Beyond its technical functionalities, the application's impact will be measured by the community it fosters and the transactions it facilitates.

As we conclude this project, our appreciation goes out to all those who have supported us on this endeavor, including mentors, peers, and individuals who have provided valuable insights. While this chapter may be closing, it marks the commencement of a new phase as the application continues to evolve, reshaping the landscape of car buying and selling through innovative digital solutions.

FUTURE ENHANCEMENTS

FUTURE ENHANCEMENTS

- ❖ In the future, this program can be enhanced by converting it into an independent application that can run without terminal or python shell.

- ❖ Features to help the customer provide ratings and reviews can also be added in the future enhancements. This will help in more interaction and user-friendliness.

- ❖ Admin control can be added to keep track of the things happening the application.

- ❖ A chat system to communicate with between the Seller and buyer therefore bringing life to the application.

BIBLIOGRAPHY

BIBLIOGRAPHY

- Computer science with Python by Sumitha Arora
- Modern Tkinter for Busy Python Developers by Mark Roseman
- https://codemy.com/intro-tkinter-python-gui-apps
- https://www.tutorialspoint.com
- https://www.geeksforgeeks.org
- https://stackoverflow.com
- https://www.javatpoint.com/python-tkinter