# "PRICE PREDICTION OF USED CARS: AN ANALYSIS"

## 1. ABSTRACT

The prediction of the price of a second-hand car is an important task in the online commerce of used vehicles. With the increasing popularity of online marketplaces for buying and selling cars, it is becoming increasingly important to understand the factors that influence the price of a used car and to develop accurate methods for predicting its price. In this paper, we examine the problem of predicting the price of a second-hand car, and we discuss the importance of collecting and analyzing a variety of data, including information on the car itself, historical sales data, economic and market data, vehicle condition and maintenance records, and location data. We also discuss the use of statistical and machine learning methods, such as linear regression, for modeling the relationship between the features of a car and its price, and for making predictions about the price of a car given its features. The results of our analysis show that linear regression can be a useful tool for predicting the price of a secondhand car, and that the accuracy of the predictions will depend on the quality and availability of the data, as well as the complexity of the model used. However, we also find that it is important to carefully evaluate the results of any predictions, and to adjust the model if necessary to improve its accuracy. Overall, this paper provides valuable insights into the problem of predicting the price of a second-hand car, and it highlights the importance of collecting and analyzing a comprehensive and diverse set of data in order to make accurate predictions

**KEYWORDS :**

- car price
- machine learning,
- Gradient Boosting.
- Artificial Intelligence API

# 2. INTRODUCTION

## 2.1 PROBLEM STATEMENT :

The problem of determining the optimal price for used cars is a significant one for both sellers and buyers. For sellers, pricing their vehicles too high can lead to difficulty in finding buyers, as most buyers will compare the price of the used car with that of a brand new one. On the other hand, pricing their vehicles too low can lead to a loss in revenue for the seller, as the car may be sold at a price lower than its actual worth.

The problem is further compounded by the fact that the pricing of used cars is influenced by a variety of factors, including the age of the vehicle, its mileage, make and model, condition, and location. These factors can vary greatly from one car to another, making it challenging to accurately determine the optimal price for a used car.

Additionally, the used car market has been growing rapidly in recent years, with more and more consumers looking to purchase pre-owned vehicles. This has led to increased competition among sellers, further highlighting the importance of pricing used cars accurately.

Therefore, the objective of this research is to develop a predictive model that accurately estimates the price of used cars, taking into account the various factors that influence the pricing. The model can provide sellers with a reliable tool to determine the optimal price for their vehicles, and for buyers to make informed purchasing decisions based on the estimated value of the used car.

## 2.2 LITERATURE REVIEW/DESCRIPTION OF PRESENT SYSTEM

Numerous studies have been conducted in the field of price prediction for used cars using statistical and machine learning models. In their study, Huang et al. (2010) proposed a hybrid model that combined a genetic algorithm with a support vector regression model to predict the prices of used cars. They found that the hybrid model outperformed traditional regression models in terms of prediction accuracy.

Similarly, Sharma et al. (2012) used a decision tree algorithm to predict the prices of used cars, taking into account various factors such as age, mileage, make, model, and location. Their study found that the decision tree algorithm was effective in predicting the prices of used cars, achieving an accuracy rate of over 80%.

Another study by Chen et al. (2013) compared the performance of different machine learning algorithms in predicting the prices of used cars. They found that the support vector regression model and the decision tree algorithm outperformed other models, achieving prediction accuracy rates of over 80%.

Currently, many online marketplaces and car dealerships use pricing algorithms to determine the optimal price for used cars. These algorithms typically take into account various factors such as age, mileage, make, model, condition, and location to estimate the value of the vehicle. However, the accuracy of these algorithms can vary widely depending on the quality and quantity of the data used, and the complexity of the algorithm used.

In general, the present system for determining the price of used cars relies on a combination of market analysis, expert opinion, and data analysis. The seller or dealer will typically consider the factors that influence the pricing of used cars, such as age, mileage, make, and model, and compare the vehicle to

other similar vehicles in the market to determine a competitive price. However, this approach can be subjective and may not take into account all the relevant factors that influence the pricing of used cars.

Therefore, there is a need for a more objective and accurate pricing model that can take into account all the relevant factors that influence the pricing of used cars. A predictive model based on statistical and machine learning algorithms can provide a more reliable tool for sellers to determine the optimal price for their vehicles, and for buyers to make informed purchasing decisions based on the estimated value of the used car.

## 2.3 BACKGROUND /LIMITATIONS

Background:

The used car market in India has been growing steadily in recent years, driven by factors such as rising income levels, increased access to credit, and a growing preference for pre-owned vehicles. According to a report by IndianBlueBook, the used car market in India is expected to reach 7 million units by 2022, up from 4 million units in 2017.

However, pricing used cars accurately remains a challenge in India, as the market is highly fragmented and lacks a standardized pricing system. Sellers and buyers often rely on subjective assessments of the value of the vehicle, based on factors such as age, mileage, make, model, and condition, and market demand.

Limitations:

One of the main limitations of price prediction models for used cars in India is the availability and quality of data. While there is some data available on the pricing of used cars, it is often incomplete, inconsistent, and outdated, making it difficult to develop accurate predictive models.

Another limitation is the variability of the used car market in India. Prices of used cars can vary widely depending on factors such as location, regional preferences, and demand for specific models. This variability makes it challenging to develop a pricing model that can accurately predict prices across different regions and segments of the market.

Additionally, there is a lack of trust and transparency in the used car market in India, which can further complicate pricing decisions. Sellers may overprice their vehicles to maximize their profits, while buyers may be hesitant to pay the asking price due to concerns about the condition of the vehicle or the accuracy of the pricing.

Finally, there is a need for greater awareness and education among sellers and buyers about the factors that influence the pricing of used cars, and the importance of objective and accurate pricing models. A more standardized approach to pricing used cars could help to improve transparency and trust in the market, and enable buyers and sellers to make more informed pricing decisions.

## 2.4 AIM & OBJECTIVES

The aim of this research paper is to develop a predictive model for estimating the price of used cars in India, taking into account various factors that influence the pricing.

The specific objectives of this research paper are:

- To identify the key factors that influence the pricing of used cars in India, such as age, mileage, make, model, condition, and location.
- To analyze the existing pricing models for used cars in India, and identify their strengths and limitations.
- To develop a predictive model for estimating the price of used cars in India, based on a comprehensive analysis of the key factors and their impact on pricing.
- To validate the accuracy of the predictive model using real-world data on used car prices in India.
- To assess the potential benefits of the predictive model for various stakeholders, such as sellers, buyers, and the used car market as a whole.
- To provide recommendations for improving the accuracy and effectiveness of pricing models for used cars in India, based on the findings of the research.

## 2.5 PROJECT MOTIVATION

The motivation behind this research project is the growing importance of the used car market in India and the need for a reliable and accurate pricing system. With the increasing demand for pre-owned vehicles, it is becoming more critical for sellers and buyers to have access to objective and reliable information about the value of used cars. A predictive model that can estimate the price of used cars accurately can help to improve transparency and trust in the market, enable sellers to optimize their pricing strategies, and help buyers make more informed decisions.

Moreover, the lack of a standardized pricing system and the fragmented nature of the used car market in India create significant challenges for both sellers and buyers. A reliable pricing model could help to address these challenges by providing a more objective and consistent approach to pricing used cars, regardless of the location or segment of the market.

Finally, this research project aims to contribute to the broader field of predictive analytics and machine learning, by developing a model that can effectively predict the price of used cars in India. The findings of the research could have implications beyond the used car market, by providing insights into the factors that influence pricing in other markets and sectors.

## 3. DESCRIPTION OF PROPOSED WORK

The user will ask to fill details of cars like Mileage, VIN, Make, Model, Year, State and City. A result will be generated at the end where the artificial intelligent will predict the price of the car.

### (3.1) Number of Modules.

- Predication model
  This module takes inputs from the seller who is selling his/her car. After filling details predicted value of car is generated.

- Buyer Model
  In this model user will buy second hand cars

- Seller Model
  In this model user will sell there cars .

## (3.2) ALGORITHM

1. Data preparation: Gather data for used cars, including features such as mileage, age, make, model, etc. Collect data on the selling price of these cars.

2. Data cleaning: Clean the data, remove missing or irrelevant data points, and convert categorical data into numerical values.

3. Data visualization: Plot the data points to visualize the relationship between the independent and dependent variables.

4. Split the data: Divide the data into training and testing sets.

5. Model building: Choose a linear regression model that suits your data. A multiple linear regression model is suitable for this project since it has more than one independent variable.

6. Model fitting: Use the training data to estimate the parameters of the regression model that best fit the data.

7. Model evaluation: Evaluate the quality of the model by analyzing the residuals (the differences between the predicted and actual values) on the testing data. If the residuals are small and randomly distributed around zero, then the model is a good fit for the data.

8. Prediction: Once you have built and evaluated the model, you can use it to predict the selling price of new used cars based on their features.

## 3.3 WORKING

"PRICE PREDICTION OF USED CARS: AN ANALYSIS" is a statistical model that attempts to predict the price of a used car based on various factors such as its age, mileage, brand, model, and other features.

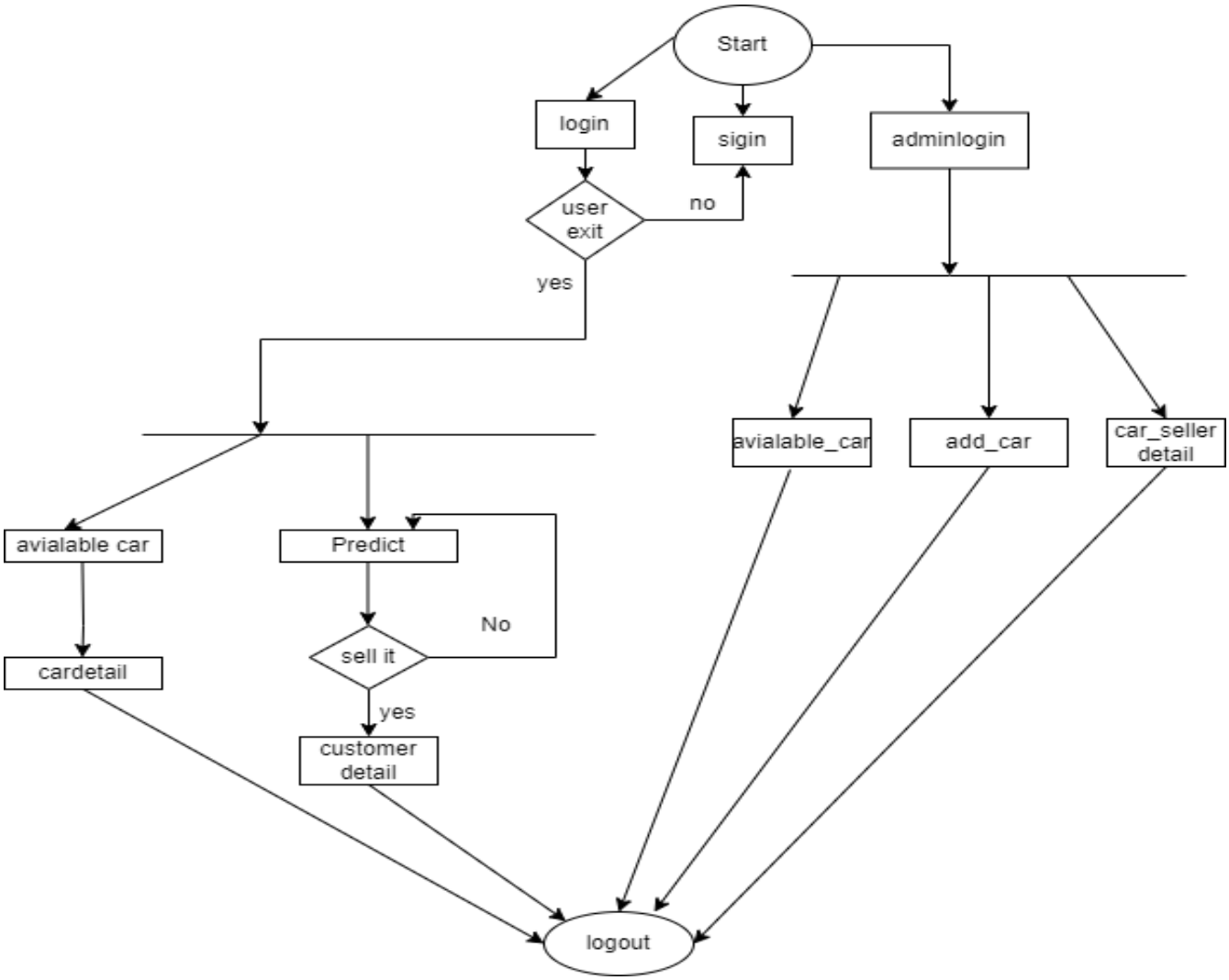
The model uses a dataset of previously sold used cars and their prices to train itself and learn the patterns and relationships between the various features and the price. This process is called supervised learning, where the model is given a labeled dataset to learn from.

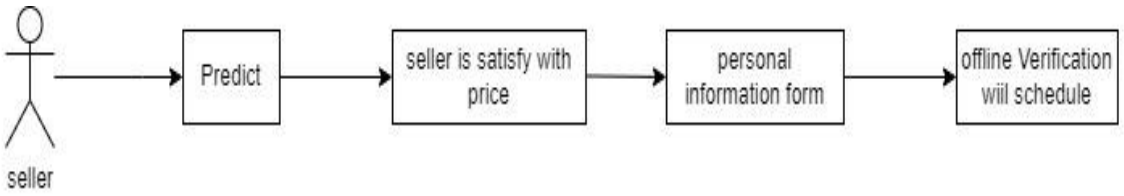
Once the model is trained, it can be used to make predictions on new or unseen data. In the case of predicting the price of a used car, the model would take in the relevant features of the car and output a predicted price.

The accuracy of the model depends on several factors such as the quality and quantity of the training data, the complexity of the model, and the features used for prediction.

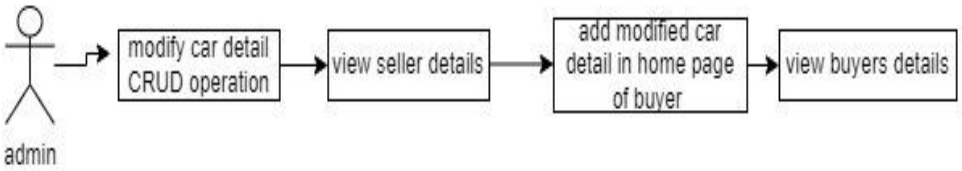
Overall, "PRICE PREDICTION OF USED CARS: AN ANALYSIS" is a useful tool for buyers and sellers of used cars to estimate the fair market value of a vehicle based on its features and characteristics.

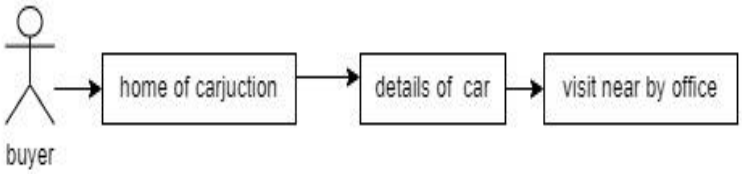
## 3.4 FLOW CHART



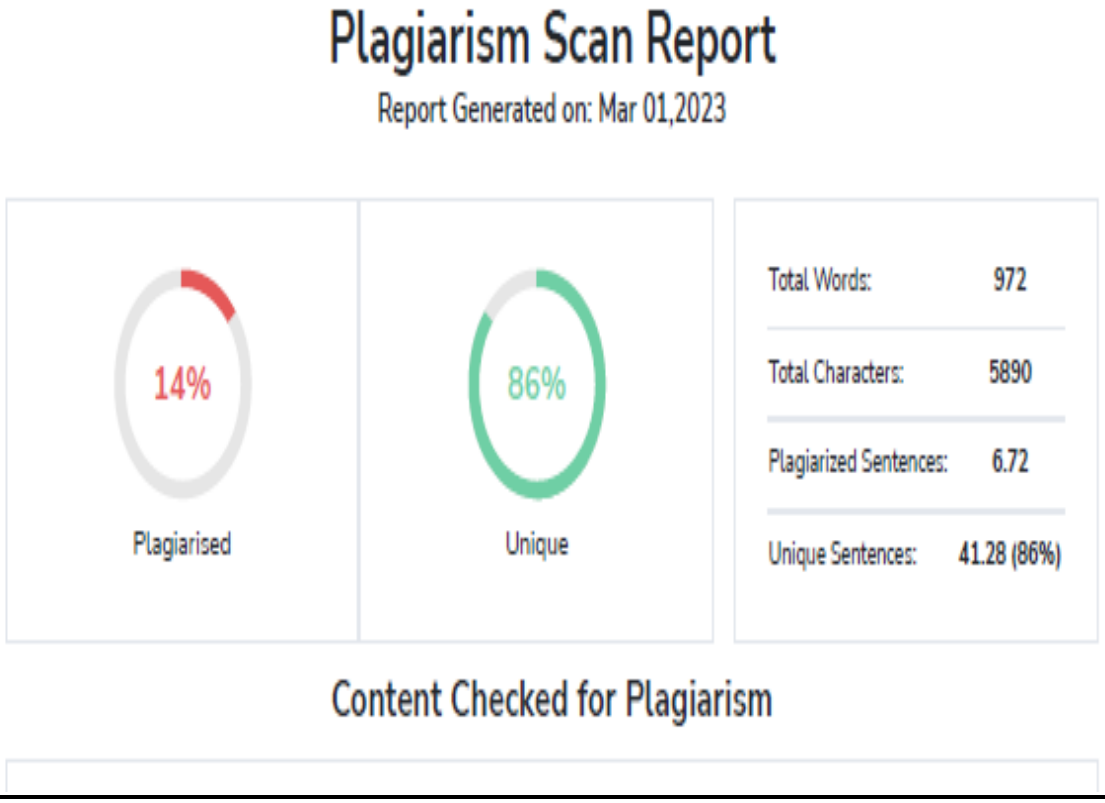

seller
Predict
seller is satisfy with price
personal information form
offline Verification wiil schedule


techical staff
offline verfication of document and car
document transfer process
seller will recieves the payment
modification of car
new price of car will decided by expert


admin
modify car detail CRUD operation
view seller details
add modified car detail in home page of buyer
view buyers details


buyer
home of carjuction
details of car
visit near by office

**3.5 Plagiarism report**




---

**3.6 Coding**

```
from werkzeug.utils import secure_filename
from werkzeug.datastructures import FileStorage
from flask import Blueprint,render_template,request,flash,g,redirect, session, url_for
from .models import *
from .models import Product
from .models import Custsell
from werkzeug.security import generate_password_hash, check_password_hash
from . import db
#from website import photos
from flask_login import login_user, login_required, logout_user, current_user
from flask_cors import CORS,cross_origin
from flask_login import login_user, login_required, logout_user, current_user
from adminproduct import ProductForm
from editproduct import EditForm
from sqlalchemy.ext.declarative import declarative_base

import os
from PIL import Image

from flask_sqlalchemy import SQLAlchemy

import secrets

import re
```

```
import pickle
import pandas as pd
import numpy.core.multiarray
import numpy as np
import sklearn

Base = declarative_base()#base directory for storing images
model=pickle.load(open('./website/LinearRegressionModel.pkl','rb'))
car=pd.read_csv('./website/Cleaned_Car_data.csv')
eregex = r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'

name_pattern = re.compile(r"^[a-zA-Z]+((['. -][a-zA-Z ])?[a-zA-Z]*)*$")

# Regular expression pattern for a phone number
phone_pattern = re.compile(r'^\+?1?\d{9,15}$')
auth = Blueprint('auth',__name__)
@auth.route('/login', methods=['GET','POST'])
def login():
    if request.method == 'POST':
        email = request.form.get('Semail')
        password = request.form.get('Spassword')

        user = User.query.filter_by(email=email).first()
        if user:
            if check_password_hash(user.password, password):
                flash('Logged in successfully!', category='success')
                login_user(user, remember=True)
                return redirect(url_for('views.home'))
            else:
                flash('Incorrect password, try again.', category='error')
        else:
            flash('Email does not exist.', category='error')
    return render_template("login.html",user=current_user)

@auth.route('/adminlogin', methods=['GET','POST'])
def adminlogin():
    if request.method == 'POST':
        username = request.form.get('Semail')
        password = request.form.get('Spassword')
        if username=="Naresh9890" and password=="9890460253":
            flash('Logged in successfully!', category='success')
            #login_user(username, remember=True)
            return redirect(url_for('auth.adminhome'))
        else:
            flash('Invalid Crediential .', category='error')
    return render_template("adminlogin.html",user=current_user)

@auth.route('/adminhome')
def adminhome():
    products = Product.query.all()
    return render_template("adminhome.html",user = current_user,products=products )
```

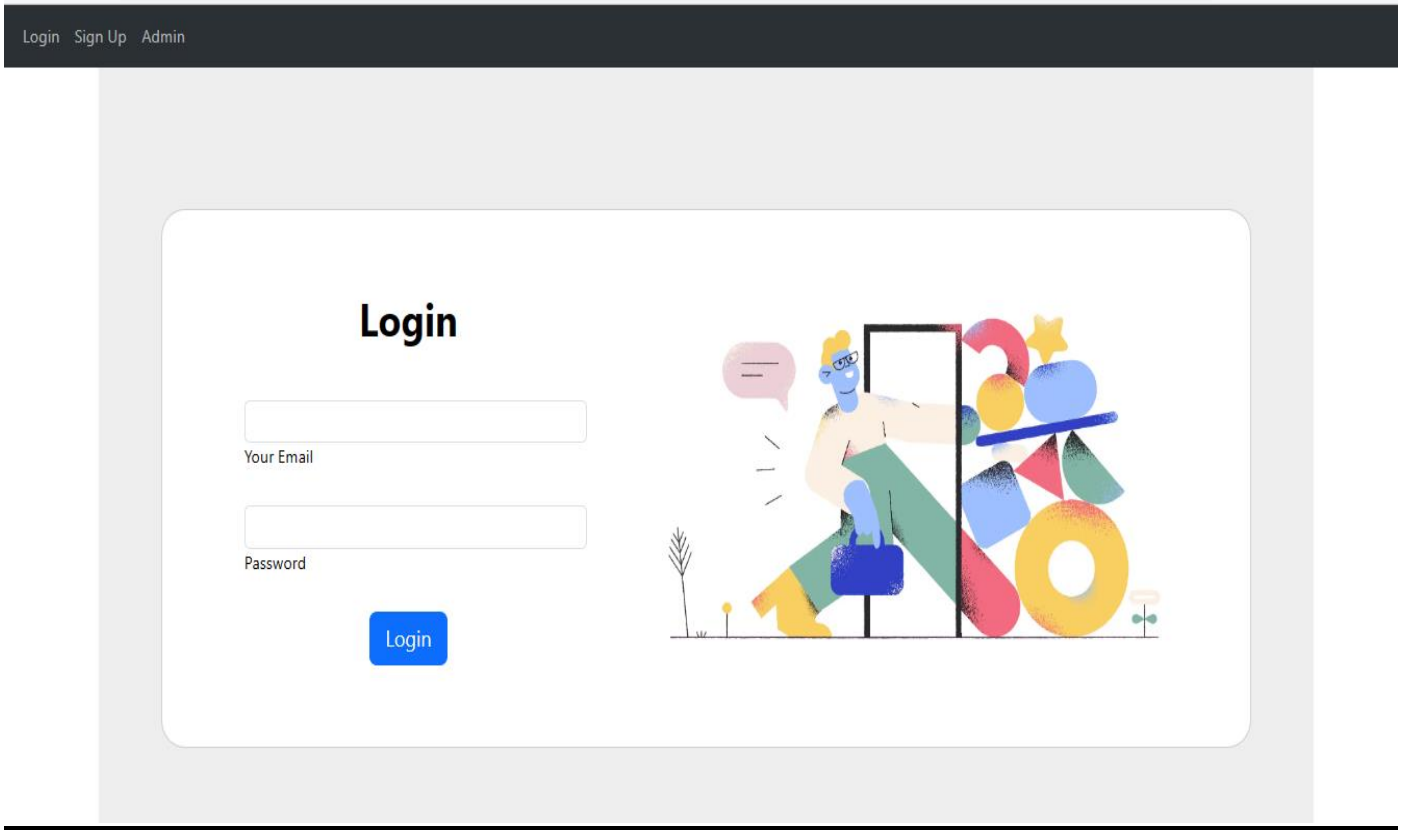
```
@auth.route('/carseller')
def carseller():
    custselling = Custsell.query.all()
    return render_template("carseller.html",user = current_user,custselling=custselling )



@auth.route('/predictioncar',methods=['GET','POST'])
def predictioncar():
    if request.method == 'POST' or 'GET':
        companies=sorted(car['company'].unique())
        car_models=sorted(car['name'].unique())
        year=sorted(car['year'].unique(),reverse=True)
        fuel_type=car['fuel_type'].unique()
        companies.insert(0,'Select Company')
        return render_template('predictioncar.html',companies=companies,
car_models=car_models, years=year,fuel_types=fuel_type,user=current_user)


@auth.route('/predict',methods=['POST'])
@cross_origin()
def predict():
    company=request.form.get('company')

    car_model=request.form.get('car_models')

    year=request.form.get('year')

    fuel_type=request.form.get('fuel_type')

    driven=request.form.get('kilo_driven')
    car = {'company':company, 'car_model':car_model, 'year':year, 'fuel_type':fuel_type,
'driven':driven}
    session['company'] = company
    session['car_model']=car_model
    session['year']=year
    session['fuel_type']=fuel_type
    session['driven']=driven
    print(car_model,company,year,driven,fuel_type)


    prediction=model.predict(pd.DataFrame(
[[car_model,company,year,driven,fuel_type]],columns=['name', 'company', 'year', 'kms_driven',
'fuel_type']))
    print(prediction)
    session['prediction']=str(np.round(prediction[0],2))
    return str(np.round(prediction[0],2))
@auth.route('/logout')
@login_required
def logout():
    logout_user()
    return redirect(url_for('auth.login'))
```

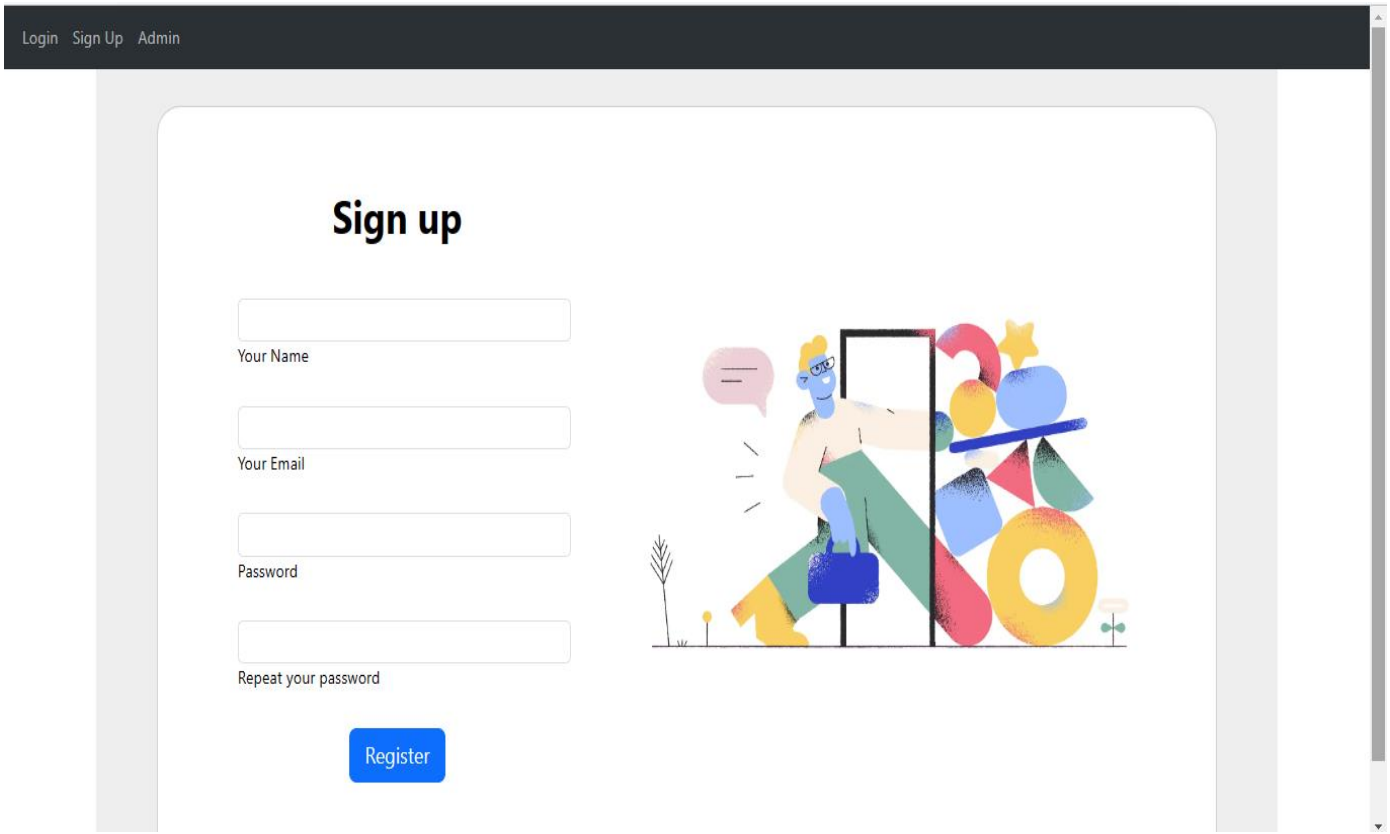
```
@auth.route('/sign_up',methods=['GET','POST'])
def sign_up():
    if request.method == 'POST':
        email = request.form.get('Semail')
        firstName = request.form.get('Sname')
        password1 = request.form.get('Spassword1')
        password2 = request.form.get('Spassword2')

        user = User.query.filter_by(email=email).first()
        if user:
            flash('Email already exists.', category='error')
        elif(False ==(re.fullmatch(eregex, email))):
            flash("Please enter the correct email",category='error')
        elif len(firstName)<3:
            flash("Please enter the correct name",category='error')
        elif len(password1)<7:
            flash("Password must be at least 7 character",category='error')
        elif password1 != password2:
            flash("Password don\'t match",category='error')
        else:
            new_user = User(email=email,first_name=firstName,
password=generate_password_hash(
                password1, method='sha256'))
            db.session.add(new_user)
            db.session.commit()
            login_user(new_user, remember=True)
            flash('Account created!', category='success')
            return redirect(url_for('auth.login'))
            flash("Account created! ", category='success')
    return render_template("sign_up.html",user=current_user)
@auth.route('/addproducts', methods=['POST','GET'] )
def addproducts():
    form = ProductForm(request.form)
    if request.method=="POST":
        name = form.name.data
        model=form.model.data
        year=form.year.data
        price=form.price.data
        description=form.description.data
        image1=(request.files.get("image1"))
        print(type(image1))
        #image1=form.image1.data

        print(image1)

        filename = secure_filename(image1.filename)
        image1.save(os.path.join("website/static/images", filename))
        if not image1:
         return "No file selected"
        #filename = secure_filename(image1.filename)
        #image1.save(os.path.join("static/images", filename))
```

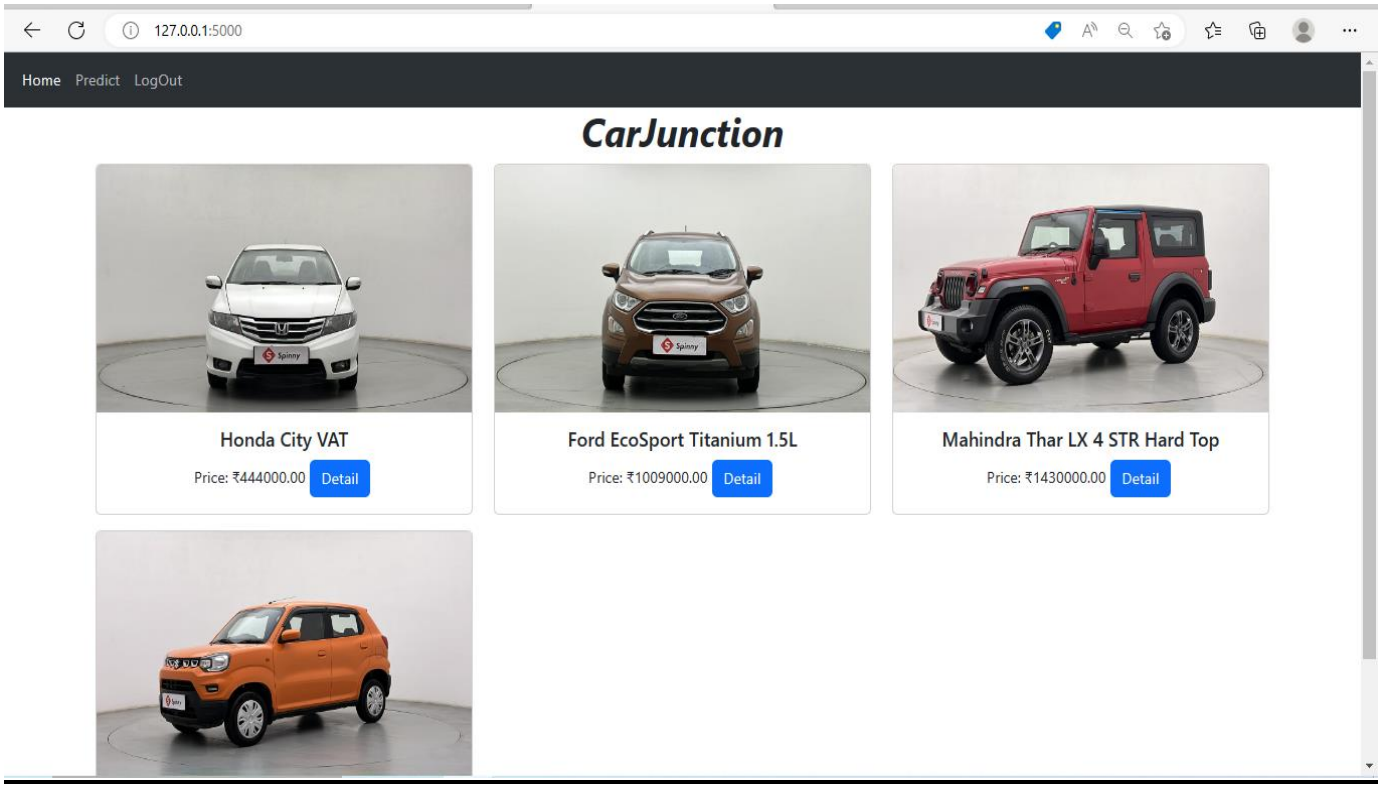
```
        #image2=photos.save(request.files.get("image2"),name=secrets.token_hex(10)+".")
        image2=request.files.get("image2")
        #image3=photos.save(request.files.get("image3"),name=secrets.token_hex(10)+".")
        filename2 = secure_filename(image2.filename)
        image2.save(os.path.join("website/static/images", filename2))
        image3=request.files.get("image3")
        filename3 = secure_filename(image3.filename)
        image3.save(os.path.join("website/static/images", filename3))
        addp                                                                    =
Product(com=name,mod=model,year=year,price=price,desc=description,img1=filename,img2=fi
lename2,img3=filename3)
        db.session.add(addp)
        db.session.commit()
        flash(f'Company {name} added successfully')
        return redirect(url_for('auth.adminhome'))
    return render_template("addproducts.html",title='add product',form=form)
@auth.route("/delete_product/<int:product_id>")
def delete_product(product_id):
    product = Product.query.get(product_id)
    db.session.delete(product)
    db.session.commit()
    return redirect("/adminhome")
@auth.route("/delete_sell/<int:sell_id>")
def delete_sell(sell_id):
    sell = Custsell.query.get(sell_id)
    db.session.delete(sell)
    db.session.commit()
    return redirect("/carseller")
@auth.route("/editproduct/<int:product_id>", methods=["GET", "POST"])
def editproduct(product_id):
    eprod = Product.query.get(product_id)
    form = EditForm(request.form)
    if request.method=="POST":
        eprod.com = form.name.data
        eprod.mod=form.model.data
        eprod.year=form.year.data
        eprod.price=form.price.data
        eprod.desc=form.description.data
        #image1=photos.save(request.files.get("image1"),name=secrets.token_hex(10)+".")
        eprod.img1=form.image1.data
        db.session.commit()
        flash(f'Product {product_id} update successfully ')
        return redirect(url_for('auth.adminhome'))
    form.name.data = eprod.com
    form.model.data = eprod.mod
    form.year.data = eprod.year
    form.price.data = eprod.price
    form.description.data = eprod.desc
    form.image1.data = eprod.img1
    return render_template('editproduct.html', form=form)
@auth.route('/customerselling', methods=["GET", "POST"])
def customerselling():
    company = session.get('company', None)
```

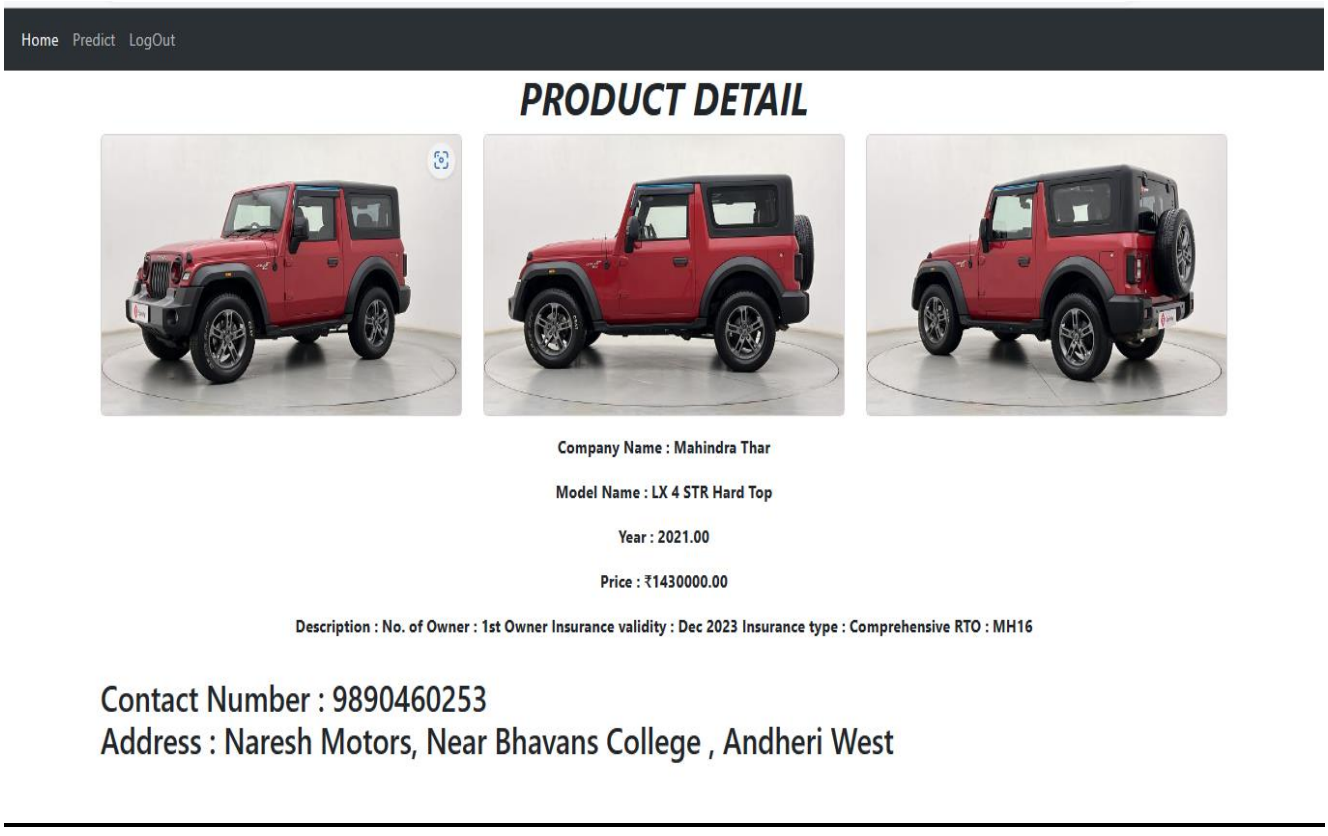
```
    model = session.get('car_model', None)
    year = session.get('year', None)
    fuel = session.get('fuel_type', None)
    driven = session.get('driven',None)
    prediction = session.get('prediction',None)
    if request.method == 'POST':
        name = request.form.get('fname')
        phone = request.form.get('phone')
        address = request.form.get('address')
        cus = Custsell.query.filter_by(phone=phone).first()
        if cus:
            flash('Phone number already exists.', category='error')
        elif not re.match(name_pattern, name):
            return flash("Please enter the correct name",category='error')
        elif not re.match(name_pattern, name):
            return flash("Please enter the correct last name",category='error')
        elif not re.match(phone_pattern, phone):
            return flash("Please enter the correct phone number",category='error')
        else:
            cus_sell =
Custsell(phone=phone,name=name,address=address,price=session.get('prediction',None))
            db.session.add(cus_sell)
            db.session.commit()
            flash('we will call you soon ', category='success')
            return redirect(url_for('auth.predictioncar'))
```

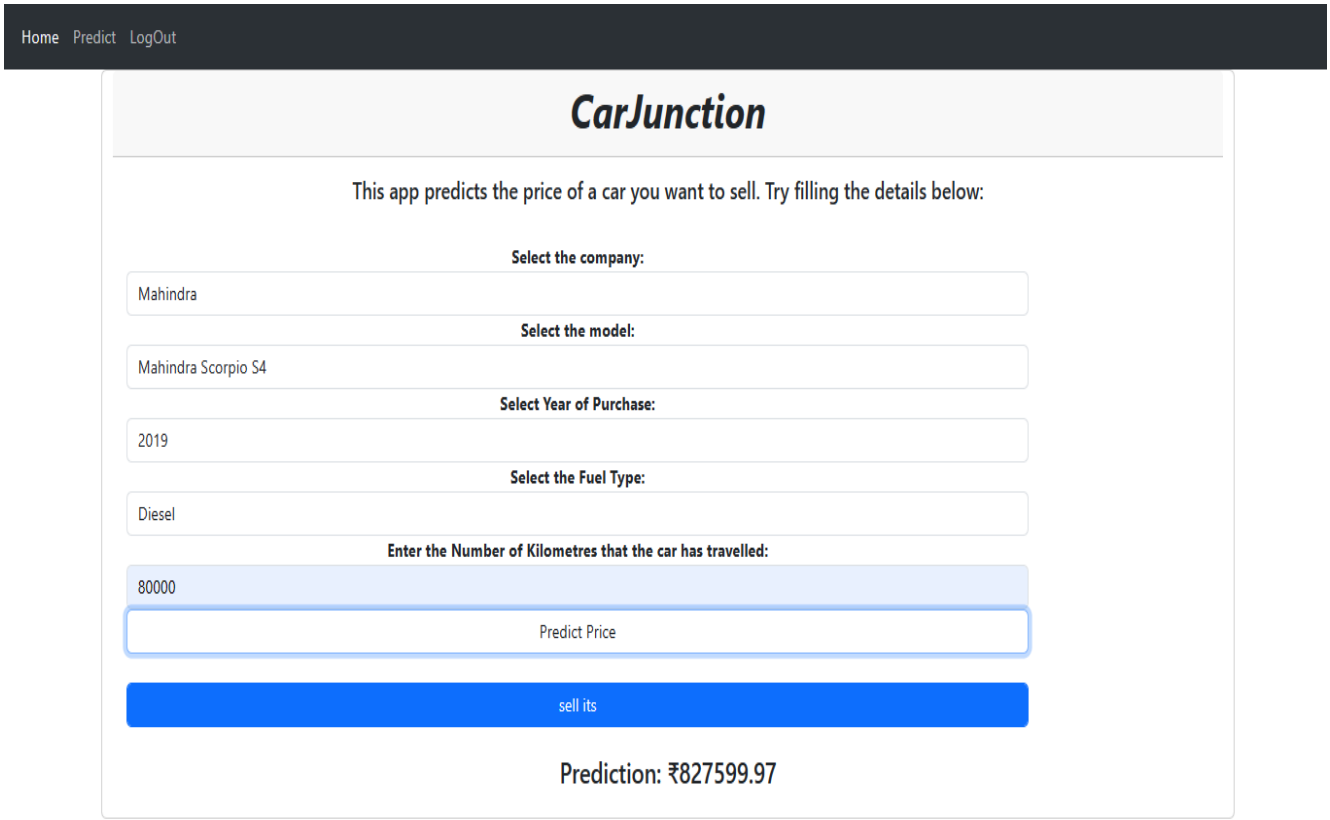
## 3.7 Screen Layouts :

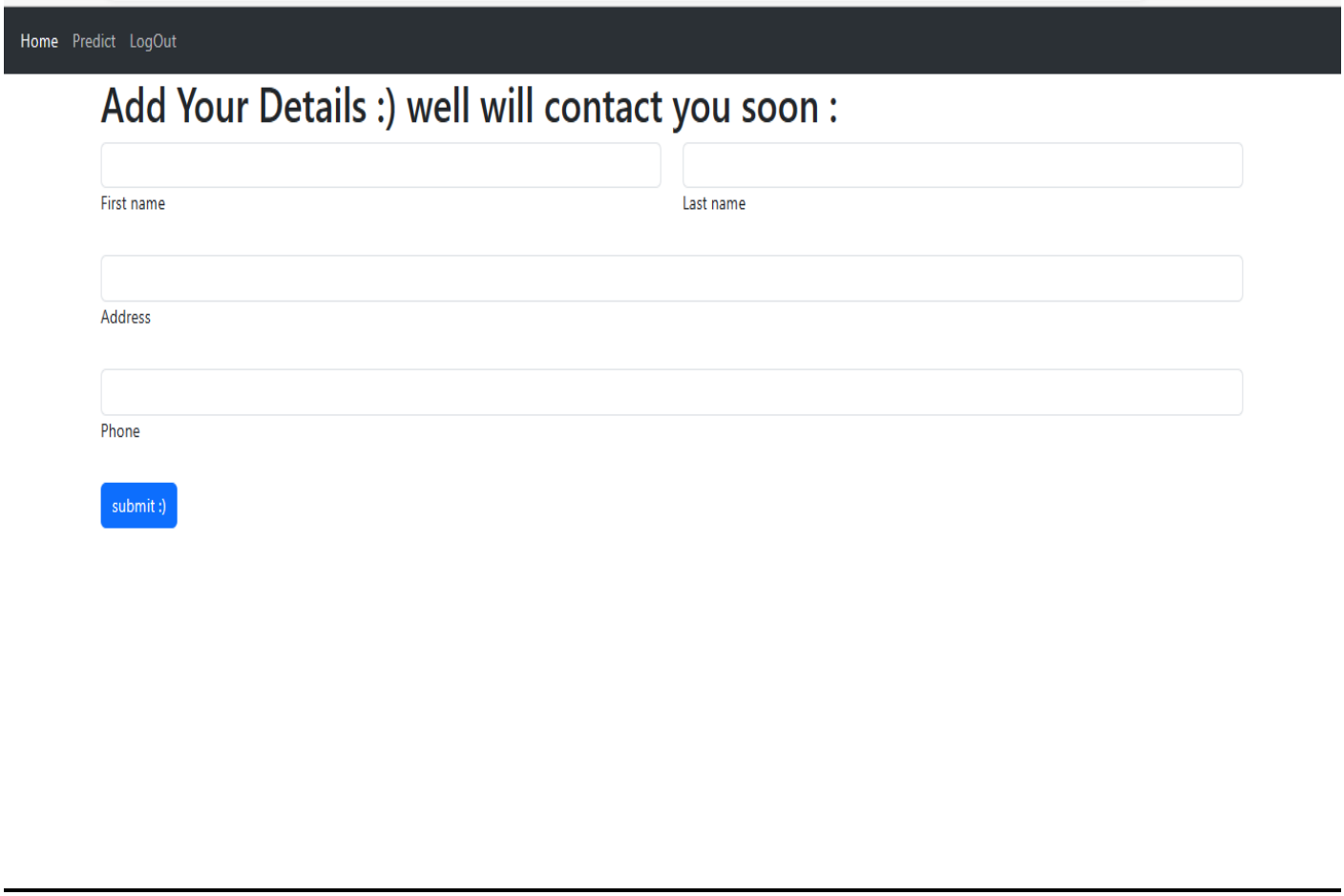



127.0.0.1:5000
Home Predict LogOut
CarJunction
Honda City VAT
Price: ₹444000.00 Detail
Ford EcoSport Titanium 1.5L
Price: ₹1009000.00 Detail
Mahindra Thar LX 4 STR Hard Top
Price: ₹1430000.00 Detail


Home Predict LogOut
PRODUCT DETAIL
Company Name : Mahindra Thar
Model Name : LX 4 STR Hard Top
Year : 2021.00
Price : ₹1430000.00
Description : No. of Owner : 1st Owner Insurance validity : Dec 2023 Insurance type : Comprehensive RTO : MH16
Contact Number : 9890460253
Address : Naresh Motors, Near Bhavans College , Andheri West


Home Predict LogOut
CarJunction
This app predicts the price of a car you want to sell. Try filling the details below:
Select the company:
Mahindra
Select the model:
Mahindra Scorpio S4
Select Year of Purchase:
2019
Select the Fuel Type:
Diesel
Enter the Number of Kilometres that the car has travelled:
80000
Predict Price
sell its
Prediction: ₹827599.97


Home Predict LogOut
Add Your Details :) well will contact you soon :
First name
Last name
Address
Phone
submit :)

AdminHome Add products Car_On_Sell LogOut

# CarJunction

< < < <

| ID | COMPANY | MODEL | YEAR | PRICE | IMAGES | ACTIONS |
|---|---|---|---|---|---|---|
| 1 | Honda City | VAT | 208.00 | 444000.00 |  | Edit Delete |
| 2 | Ford EcoSport | Titanium 1.5L | 2021.00 | 1009000.00 |  | Edit Delete |
| 3 | Mahindra Thar | LX 4 STR Hard Top | 2021.00 | 1430000.00 |  | Edit Delete |
| 4 | Maruti Suzuki | S-Presso VXI+ | 2019.00 | 475000.00 |  | Edit Delete |

AdminHome Add products Car_On_Sell LogOut

## Add a product

**company**

Add company name

**model**

Add model name

**year**

Add manufacture year

**Price**

Add product price

**Description**

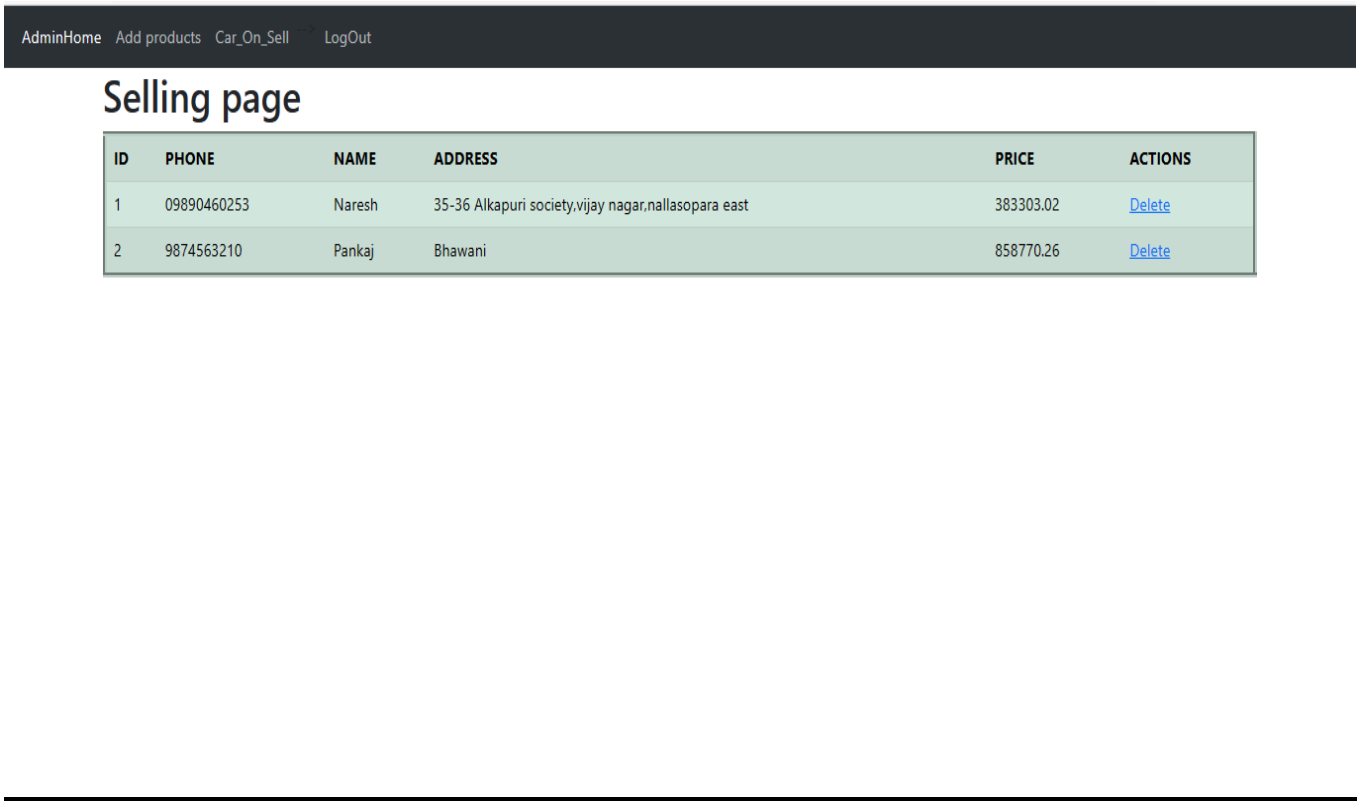
Add product name

**Image** Choose File No file chosen

**Image** Choose File No file chosen

**Image** Choose File No file chosen

Add Product


AdminHome Add products Car_On_Sell LogOut
Selling page
ID PHONE NAME ADDRESS PRICE ACTIONS
1 09890460253 Naresh 35-36 Alkapuri society,vijay nagar,nallasopara east 383303.02 Delete
2 9874563210 Pankaj Bhawani 858770.26 Delete

## 4 TECHNOLOGY/LANGUAGE/DEVELOPMENT TOOLS/HARDWARE

- Html
- CSS
- Bootstrap
- Artificial Intelligence API
- Javascript
- Php
- Flask Framework
- Python library such as Numpy ,Scipy ,Scikit-Learn and Jupyter Notebook etc

## HARDWARE

- Requires a 2.6 GHz or faster processor. Quad core or better recommended
- Requires 4 GB of RAM or higher for load generation
- Requires 10 GB of hard disk space

## 5 CONCLUSION & FUTURE SCOPE

The prediction of the price of a car is a complex task that involves the collection and analysis of a variety of data, including information on the car itself, historical sales data, economic and market data, vehicle condition and maintenance records, and location data. Statistical and machine learning methods, such as linear regression, can be used to model the relationship between the features of a car and its price, and to make predictions about the price of a car given its features. While linear regression can be a useful tool for predicting car prices, it is important to note that the accuracy of the predictions will depend on the quality and availability of the data, as well as the complexity of the model used. As such, it is important to carefully evaluate the results of any predictions, and to adjust the model if necessary to improve its accuracy. Overall, the goal of predicting the price of a car is to provide valuable insights into the automotive market, and to inform decision-making for both buyers and sellers of second-hand cars

## 6 References/Resource Material/Data collection

REFERENCES WEBSITES:


- http://www.google.com/
- https://www.kaggle.com