EWRJLKHFKL

A Major Project

On

**CAR PRICE PREDICTION USING MACHINE LEARNING**

(Submitted in partial fulfilment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

**B. Sharath Chandra Reddy (19E41A05K4)**

**CH. Karthik (19E41A05G2)**

**Y. Akshith (19E41A05H9)**

**AC.Adarsh (19E41A05M0)**

Under the Guidance of

Mrs. Deepthi Pola

###### (Assistant Professor)



###### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**SREE DATTHA INSTITUTE OF ENGINEERING AND SCIENCE**

(Accredited by NAAC, Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under Section 2(f) of the UGCAct.1956,

Sheriguda (V), Ibrahimpatnam (M), Ranga Reddy – 501510.

**2019-2023**

EWRJLKHFKL

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**CERTIFICATE**

###### This is to certify that the project entitled “CAR PRICE PREDICTION USING MACHINE LEARNING” is being submitted by B.SHARATH CHANDRA REDDY (19E41A05K4), CH.KARTHIK(19E41A05G2),Y.AKSHITH(19E41A05H9),AC.ADARSH(19E41A05M0) in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by him/her under our guidance and supervision during the year 2022-23.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

|  |  |
| --- | --- |
| **Deepthi Pola** | **Dr. P. Srinivasa Rao** |
| **(Internal Guide)** | **(Principal)** |
| **Dr. S VenkataAchuta Rao**  ( **HOD-CSE & Dean- Academics)** | **(External Examiner)** |

**Submitted for viva voice Examination held on**

###### ACKNOWLEDGEMENT

Apart from our efforts, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard for my internal guide **Mrs. DEEPTHI POLA,** Assistant Professor for her exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We are also thankful to **Dr. S Venkata Achuta Rao,** Dean Academicsand Head of the Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. P. Srinivasa Rao,** Principal for being cooperative throughout this project. We would like to express our sincere gratitude to **Dr. GNV Vibhav Reddy**, Vice Chairman and **Sri. G. Panduranga Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout this project.

The guidance and support were received from all the members of **Sree Dattha Institute of Engineering and Science** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

**ABSTRACT**

The charge of a brand new vehicle with inside the enterprise is constant via way of means of the producer with a few extra expenses incurred via way of means of the Government with inside the shape of taxes. So, clients shopping for a brand new vehicle may be confident of the cash they make investments to be worthy. But, because of the expanded fees of latest automobiles and the financial inability of the clients to shop for them, Used Car income are on an international increase. Therefore, there may be an pressing want for a Used Car Price Prediction gadget which efficaciously determines the worthiness of the auto the use of quite a few features. In this paper, we check out the software of supervised device learning strategies to are expecting the charge of used automobiles. Different strategies (Regression Algorithms) are used to make the predictions. The predictions are then evaluated and in comparison on the way to discover the ones which offer the best performances.

**Keywords:** Car Price Prediction, Linear Regression, Machine Learning, dependent variable etc.

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **FIGURE NAME** | **PAGE NO** |
| Figure 4.1 | Project Architecture | 35 |
| Figure 4.2 | Use case diagram | 37 |
| Figure 4.3 | Class diagram | 38 |
| Figure 4.4 | Sequence diagram | 39 |
| Figure 4.5 | Activity diagram | 40 |

**LIST OF SCREENSHOTS**

|  |  |  |
| --- | --- | --- |
| **SCREENSHOT NO.** | **SCREENSHOT NAME** | **PAGE NO** |
| Screenshot 6.1 | LAYOUT OF TESTING PLATFROM | 53 |
| Screenshot 6.2 | GO TO THE GOOGLE AND PAST THE HOST LOCATION | 54 |
| Screenshot 6.3 | THIS THE HOME WEB PAGE | 54 |
| Screenshot 6.4 | REGISTER PAGE OF WEB APPLICATION | 55 |
| Screenshot 6.5 | LOGIN PAGE OF WEB APPLICATION | 55 |
| Screenshot 6.6 | HOME PAGE OF WEB APPLICATION | 56 |
| Screenshot 6.7 | DISPLAYING AVAILABLE CAR COMPANIES | 56 |
| Screenshot 6.8 | DISPLAYING SUITABLE CAR MODEL | 57 |
| Screenshot 6.9 | DISPLAYING AVAILABLE YEARS | 57 |
| Screenshot 6.10 | DISPALYING AVAILABLE FUEL TYPE | 58 |
| Screenshot 6.11 | ENTER THE KILOMETERS TRAVELLED | 58 |
| Screenshot 6.12 | DISPLAYING PREDICTED PRICE | 59 |

**TABLE OF CONTENTS**

**ABSTRACT**  I

**LIST OF FIGURES**  II

**LIST OF SCREENSHOTS** III

1. **INTRODUCTION** 1
   1. PROJECT SCOPE 2
   2. PROJECT PURPOSE 2
   3. PROJECT FEATURES 2
2. **LITERATURE SURVEY** 3-9
3. **SYSTEM ANALYSIS** 10
   1. EXISTING SYSTEM 11

DISADVANTAGES

* 1. PROPOSED SYSTEM 12

ADVANTAGES

* 1. FEASIBILITY STUDY 12
     1. ECONOMIC FESIBILITY 13
     2. TECHNICAL FEASIBILITY 13
     3. SOCIAL FEASIBILITY 13
  2. SYSTEM REQUIREMENTS 14
     1. HARDWARE REQUIREMENTS 14
     2. SOFTWARE REQUIREMENTS 14
     3. TECHNOLOGIES USED 15-33

1. **ARCHITECTURE** 34-35
   1. UML DIAGRAMS 36
      1. USE CASE DIAGRAM 37
      2. CLASS DIAGRAM 38
      3. SEQUENCE DIAGRAM 39
      4. ACTIVITY DIAGRAM 40
2. **IMPLEMENTATION** 41
   1. FRONTEND (WEBAPP. PY) 42-47
   2. BACKEND (DETECT. PY) 48-51
3. **SCREENSHOTS** 52-59
4. **TESTING**  60
   1. INTRODUCTION TO TESTING 61
   2. TYPES OF TESTING 61
      1. UNIT TESTING 61
      2. INTEGRATION TESTING 61
      3. FUNCTIONAL TESTING 62-64
   3. TEST CASES 65
5. **CONCLUSION & FUTURE SCOPE**  66
   1. PROJECT CONCLUSION 67
   2. FUTURE SCOPE 67
6. **BIBLIOGRAPHY**68
   1. GITHUB LINK 69-70
   2. REFERENCES 69-70

## 1. INTRODUCTION

###### 1. INTRODUCTION

### 1.1 PROJECT SCOPE

This project is titled as “**CAR PRICE PREDICTION USING MACHINE LEARNING**”. Predicting the resale value of a car is not a simple task. It is trite knowledge that the value of used cars depends on a number of factors. Unfortunately, in practice, most people do not know exactly how much fuel their car consumes for each km driven. Other factors such as the type of fuel it uses, the interior style, the braking reviews, prestigious awards won by the car manufacturer, its physical state. whether it is a sports car, whether it has cruise control, whether it is automatic or manual transmission, whether it belonged to an individual or a company and other options such as air conditioner, sound system, power steering, cosmic wheels, GPS navigator all may influence the price as well.

### 1.2 PROJECT PURPOSE

### The used car market is an ever-rising industry, which has almost doubled its market value in the last few years. The emergence of online portals such as CarDheko, Quikr, Carwale, Cars24, and many others has facilitated the need for both the customer and the seller to be better informed about the trends and patterns that determine the value of the used car in the market. Machine Learning algorithms can be used to predict the retail value of a car, based on a certain set of features. The purpose of this project is to provide Car price prediction using machine learning without any human interference. In our day to day lives everyone buys and sells a car every day. Now there are limited facilities and applications to get an appropriate price for one’s car. Now we use this application to get an estimate value of the car.

### 1.3 PROJECT FEATURES

The main limitation of this study is the low number of records that have been used.As future work, we intend to collect more data and to use more advanced techniques like artificial neural networks, fuzzy logic and genetic algorithms to predict car prices.

## 2. LITERATURE SURVEY

###### 2. LITERATURE SURVEY

**2.1 Interpretation of the Correlation Coefficient: A Basic Review.**

**AUTHORS:** R.Taylor,

**ABSTRACT:** A basic consideration in the evaluation of professional medical literature is being able to understand the statistical analysis presented. One of the more frequently reported statistical methods involves correlation analysis where a correlation coefficient is reported representing the degree of linear association between two variables. This article discusses the basic aspects of correlation analysis with examples given from professional journals and focuses on the interpretations and limitations of the correlation coefficient. No attention was given to the actual calculation of this statistical value.

**2.2 Design and Development of Data Mining System to Analyze Cars using Improved ID3 with TKNN Clustering Algorithm.**

**AUTHORS:** M. Jayakameswaraiah and S. Ramakrishna

**ABSTRACT:** Conventional way of business is a challenging in car market due to many competitors are there around the world for providing competitive products. The car manufacturers categorizes the car users and have to invent a suitable car; the seller correctly groups the buyers and he sells a right car and the customers selects best car by analyzing more brands of cars with 'N' number of sellers. These three cases they spent too much of time for analyzing old or statistical data for choosing a right product. Now a day's customers are required comfort and their loving brand & color. With the advent of the Internet and Data Mining Algorithms has undoubtedly contributed to the shift of marketing focus. In this paper, we proposed Improved ID3 with TkNN algorithm for best car market analysis. We have executed the same in WEKA Tool with Java code. We analyzed the graphical performance analysis between TKNN and our novel improved ID3 with TKNN clustering algorithms with Classes to Clusters evaluation purchase, safety, luggage booting, persons (seating capacity), doors, maintenance and buying attributes of customer's requirements for unacceptable/acceptable/good/very good ratings of a car to purchase.

**2.3 Implementation of naive Bayes classification method for predicting purchase.**

**AUTHORS**: F. Harahap, A. Y. N. Harahap, E. Ekadiansyah, R. N. Sari, R. Adawiyah, and C. B. Harahap

**ABSTRACT:** To choose the right vehicle according to the needs and funds owned by consumers, requires a careful analysis that takes into account many criteria and factors. The criteria used as a benchmark in choosing a vehicle, among others, price, spare parts, cylinder volume, the power of the vehicle. To process all these criteria required a system that can select and classify criteria chosen by consumer, so that can assist consumer in choosing the most appropriate vehicle, therefore needed a system for decision making in making car purchase. The Naive Bayes algorithm is a simple probabilistic classifier that computes a set of probabilities by summing the frequency and value combinations of the given dataset. Application of Naïve Bayes method is expected to be able to predict car purchases. Of the 20 car purchase data used in the test by the Naïve Bayes method, then obtained a percentage of 75% for the accuracy of prediction, where from 20 car purchase data tested there are 15 data purchase car successfully classified correctly.

**2.4 A prediction study on the car sales based on web search data.**

**AUTHORS:** Q. Yuan, Y. Liu, G. Peng, and B. Lv

**ABSTRACT:** The automobile industry is a pillar industry in the national economy, so it's of important significance to predict accurately car sales. On the basis of building theoretical frame of web search data and car sales, this paper selects keywords with the technology of auto recommendation, composes the keywords into composite index which can be used in the regression model of this paper, analyzes the relationship between the sales of car in different price ranges and the corresponding composite index, makes the prediction test, finally has the forecast results of car sales in different price ranges whose Prediction MAPE is less than 4 percent.

**2.5 Secondhand car price estimation using artificial neural network.**

**AUTHORS:** M. C. Sorkun,

**ABSTRACT:** The objective of this study is the investigation of ANN performance on secondhand car price estimation. This study has two main steps to prepare the data and applications with ANN. The first step involves the collection and preparation of raw data for the ANN implementation. The second step involves the creation of the ANN model. The data used in this steps were collected from secondhand goods selling website. The results of this study were compared with related works.

**Table 2.1: LITERATURE REVIEW**

|  |  |  |  |
| --- | --- | --- | --- |
| **Title** | **Authors** | **Year** | **Description** |
| Interpretation of the Correlation Coefficient: A Basic Review | R.Taylor, | 25 march 2020 | In this paper, This article discusses the basic aspects of correlation analysis with examples given from professional journals and focuses on the interpretations and limitations of the correlation coefficient. |
| Design and Development of Data Mining System to Analyze Cars using Improved ID3 with TKNN Clustering Algorithm | M. Jayakameswaraiah and S. Ramakrishna | 14 September 2018 | In this paper, we proposed Improved ID3 with TkNN algorithm for best car market analysis. We have executed the same in WEKA Tool with Java code. We analyzed the graphical performance analysis between TKNN and our novel improved ID3 with TKNN clustering algorithms with Classes to Clusters evaluation purchase, safety, luggage booting, persons (seating capacity), doors, maintenance and buying attributes of customer's requirements for unacceptable/acceptable/good/very good ratings of a car to purchase. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Implementation of naive Bayes classification method for predicting purchase. | F. Harahap, A. Y. N. Harahap, E. Ekadiansyah, R. N. Sari, R. Adawiyah, and C. B. Harahap |  | 11 April 2021 | In this paper, Application of Naïve Bayes method is expected to be able to predict car purchases. Of the 20 car purchase data used in the test by the Naïve Bayes method, then obtained a percentage of 75% for the accuracy of prediction. |
| A prediction study on the car sales based on web search data. | Q. Yuan, Y. Liu, G. Peng, and B. Lv |  | 23 July 2022 | In this paper, selects keywords with the technology of auto recommendation, composes the keywords into composite index which can be used in the regression model of this paper, analyzes the relationship between the sales of car in different price ranges and the corresponding composite index, makes the prediction test, finally has the forecast results of car sales in different price ranges whose Prediction MAPE is less than 4 percent. |
| Secondhand car price estimation using artificial neural network. | M. C. Sorkun, | 7 May 2019 | | In this paper, The data used in this steps were collected from secondhand goods selling website. The results of this study were compared with related works. |

## 3. SYSTEM ANALYSIS

###### 3. SYSTEM ANALYSIS

### 3.1. EXISTING SYSTEM

##### System includes a process where a seller decides a price randomly and buyer has no idea about the car and it’s value in the present day scenario. In fact, seller also has no idea about the car’s existing value or the price he should be selling the car .To overcome this problem we have developed a model which will be highly effective Regression Algorithms are used because they provide us with continuous value as an output and not a categorized value. Because of which it will be possible to predict the actual price a car rather than the price range of a car.User Interface has also been developed which acquires input from any user and displays the Price of a car according to user’s inputs.

##### Disadvantages

##### 1) Regression algorithms provide high effective model but with less accuracy.

##### 2) Focuses on high quality predictions(single model) .

##### 3) Weak learner models .

**3.2. PROPOSED SYSTEM**

Boosting is an ensemble learning technique that uses a set of Machine Learning algorithms to convert weak learner to strong learners. In order to increase the accuracy of the model. XGBoost is a specific implementation of the Gradient Boosting method which uses more accurate approximations to find the best model. It employs a number of nifty tricks that make it exceptionally successful, particularly with structured data.

Advantages

1) Boosting Algorithms seek to improve the prediction power by training a sequence of weak models, each compensating the weaknesses of its predecessors.

2) XGBoost has additional advantages: training is very fast and can be parallelized / distributed across clusters. Therefore, XGBoost was another model that is used in this study.

3) Prediction is more accurate

4) This makes a strong learner model.

3.3. FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

3.3.1. ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

#### 3.3.2. TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

#### 3.3.3. SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

### 3.4. SYSTEM REQUIREMENTS

#### 3.4.1. HARDWARE REQUIREMENTS

#### Operating system : Windows 7 or Above.

#### Coding Language : Python.

#### Front-End : Python.

#### Back-End : Django.

#### Designing : Html, css, javascript.

#### Data Base : MySQL.

# 3.4.2. SOFTWARE REQUIREMENTS

* Processor - i3 or Above.
* RAM - 4 GB or Above.
* Hard Disk - 50 GB or Above.
* Key Board - Standard Window Keyboard.
* Mouse - Optical Mouse.
* Monitor - Monitor or Laptop.

**3.4.3. TECHNOLOGIES USED**

**PYTHON**

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system. Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked. Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting Unicode. Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it. "Some changes in Python 7.3:

• Print is now a function

• Views and iterators instead of lists

• The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.

• There is only one integer type left, i.e. int. long is int as well.

• The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.

**Purpose**

We demonstrated that our approach enables successful segmentation of intra-retinal layers— even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

**Below are some facts about Python**

Python is currently the most widely used multi-purpose, high-level programming language.Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, and Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

* [Machine Learning](https://www.geeksforgeeks.org/machine-learning/)
* GUI Applications (like Kivy, Tkinter, PyQt etc. )
* Web frameworks like Django (used by YouTube, Instagram, Dropbox)
* Image processing (like Opencv, Pillow)
* Web scraping (like Scrapy, BeautifulSoup, Selenium)
* Test frameworks
* Multimedia

**Modules Used in Project:-**

Argparse

The [argparse mo](https://docs.python.org/3/library/argparse.html#module-argparse)dule makes it easy to write user-friendly command-line interfaces. The program defines what arguments it requires, and [argparse wi](https://docs.python.org/3/library/argparse.html#module-argparse)ll figure out how to parse those out of [sys.argv. T](https://docs.python.org/3/library/sys.html#sys.argv)he [argparse mo](https://docs.python.org/3/library/argparse.html#module-argparse)dule also automatically generates help and usage messages. The module will also issue errors when users give the program invalid arguments.

**Numpy**

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated (broadcasting) functions
* Tools for integrating C/C++ and Fortran code
* Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

**Pandas**

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

**Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and [IPython sh](http://ipython.org/)ells, the [Jupyter No](http://jupyter.org/)tebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the [sample plots an](https://matplotlib.org/tutorials/introductory/sample_plots.html)[d thumbnail gallery.](https://matplotlib.org/gallery/index.html) For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

**Pillow**

The Pillow library contains all the basic image processing functionality. You can do image resizing, rotation and transformation. Pillow module allows you to pull some statistics data out of image using histogram method, which later can be used for statistical analysis and automatic contrast enhancement.

**GTTS**

gTTS (Google Text-to-Speech), a Python library and CLI tool to interface with Google Translate's text-to-speech API. Writes spoken mp3 data to a file, a file-like object (bytestring) for further audio manipulation, or stdout . It features flexible pre-processing and tokenizing.

**Torch**

Torch is an open source ML library used for creating deep neural networks and is written in the Lua scripting language. It's one of the preferred platforms for deep learning research. The framework is built to speed up the process between research prototyping and deployment.

**Flask**

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

**Advantages of Python:-**

Let’s see how Python dominates over other languages.

1. **Extensive Libraries**

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don’t have to write the complete code for that manually.

1. **Extensible**

As we have seen earlier, Python can be **extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

1. **Embeddable**

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilitiesto our code in the other language.

1. **Improved Productivity**

The language’s simplicity and extensive libraries render programmersmore productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

1. **IOT Opportunities**

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

1. **Easy to Code**

When working with Java, you may have to create a class to print **‘Hello World’**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and **code.** This is why when people pick up Python; they have a hard time adjusting to other more verbose languages like Java.

1. **Readable**

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

1. **Object-Oriented**

This language supports both the **procedural and object-oriented** programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

1. **Free and Open-Source**

Like we said earlier, Python is **freely available.** But not only can you [**download Python** fo](https://data-flair.training/blogs/install-python-windows/)r free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

1. **Portable**

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to **code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

1. **Interpreted**

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.

**Advantages of Python Over Other Languages :**

1. **Less Coding**

Almost all of the tasks done in Python require less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don’t have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

1. **Affordable**

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

**The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.**

1. **Python is for everyone**

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and [**machine learning**, a](https://data-flair.training/blogs/machine-learning-tutorials-home/)utomate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

**Disadvantages of Python**

So far, we’ve seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let’s now see the downsides of choosing Python over another language.

1. **Speed Limitations**

We have seen that Python code is executed line by line. But since [Python is](https://www.python.org/) interpreted, it often results in **slow execution**. This, however, isn’t a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

1. **Weak in Mobile Computing and Browsers**

While it serves as an excellent server-side language, Python is much rarely seen on the **clientside**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**.

The reason it is not so famous despite the existence of Brython is that it isn’t that secure.

1. **Design Restrictions**

As you know, Python is **dynamically-typed**. This means that you don’t need to declare the type of variable while writing the code. It uses **duck-typing**. But wait, what’s that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can **raise run-time errors**.

1. **Underdeveloped Database Access Layers**

Compared to more widely used technologies like **JDBC (Java DataBase**

**Connectivity)** and **ODBC (Open DataBase Connectivity)**, Python’s database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

1. **Simple**

No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

**History of Python: -**

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python.Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners1, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

**DEEP LEARNING**

Deep learning is a subfield of machine learning that focuses on training artificial neural networks to automatically learn and extract meaningful representations from data. It aims to emulate the human brain's ability to process and understand complex patterns, enabling computers to perform tasks such as image and speech recognition, natural language processing, and decision-making.

Unlike traditional machine learning approaches that rely on handcrafted features, deep learning algorithms learn hierarchical representations of data directly from raw inputs. This is accomplished through the use of deep neural networks, which consist of interconnected layers of artificial neurons. Each neuron applies a non-linear transformation to its inputs and passes the result to the next layer.

Deep learning models are trained through a process called backpropagation, which involves iteratively adjusting the weights of the neural network based on the difference between the predicted output and the actual output. This optimization process aims to minimize a predefined loss function, allowing the network to gradually improve its performance.

One key aspect of deep learning is its ability to automatically learn hierarchical feature representations. Lower layers of the network learn simple and local features, such as edges or textures, while higher layers learn more complex and abstract representations. This hierarchical representation learning enables deep learning models to capture intricate patterns and dependencies in the data.

Deep learning models typically require large amounts of labeled training data to generalize well to unseen examples. The availability of massive datasets and advancements in computational power, particularly the use of graphics processing units (GPUs), have been crucial in fueling the success of deep learning in recent years.

Another important aspect of deep learning is its flexibility in handling different types of data, including images, text, audio, and video. Deep neural networks can be adapted and customized for specific tasks and domains, allowing for a wide range of applications in various fields, including computer vision, natural language processing, healthcare, finance, and more.

In summary, deep learning is a branch of machine learning that focuses on training neural networks with multiple layers to learn hierarchical representations of data. It leverages the power of large datasets and complex architectures to automatically extract meaningful features and make predictions. Through its ability to handle diverse data types and its potential for capturing intricate patterns, deep learning has become a powerful tool in solving complex problems in artificial intelligence.

**Categories of Deep Learning**

Deep learning can be categorized into several subfields based on the specific techniques and architectures used. Here are some common categories of deep learning:

1. **Convolutional Neural Networks (CNNs):** CNNs are primarily used for image and video processing tasks. They are designed to automatically learn and extract hierarchical features from visual data. CNNs consist of convolutional layers that apply filters to capture spatial information, followed by pooling layers for downsampling and fully connected layers for classification.
2. **Recurrent Neural Networks (RNNs):** RNNs are suited for sequence data processing, such as natural language processing and speech recognition. They have recurrent connections that allow information to persist over time, making them effective in modeling sequential dependencies. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) are popular RNN variants.
3. **Generative Adversarial Networks (GANs):** GANs are used for generative modeling tasks, such as creating realistic synthetic images or generating new data samples. GANs consist of two components: a generator network that produces synthetic samples and a discriminator network that tries to differentiate between real and synthetic samples. They are trained in an adversarial manner to improve the quality of generated samples.

1. **Auto-encoders:** Auto-encoders are unsupervised learning models that aim to learn efficient representations of input data. They consist of an encoder network that compresses the input data into a lower-dimensional representation and a decoder network that reconstructs the original input from the encoded representation. Auto-encoders are used for tasks such as dimensionality reduction, anomaly detection, and de-noising.

1. **Reinforcement Learning (RL):** RL focuses on training agents to learn optimal actions in a dynamic environment through trial and error. Deep Reinforcement Learning (DRL) combines deep learning with RL algorithms to handle high-dimensional state and action spaces. DRL has been successful in game playing, robotics, and other sequential decision-making tasks.
2. **Transformers:** Transformers have gained significant attention in natural language processing tasks, particularly in machine translation and text generation. Transformers utilize selfattention mechanisms to capture dependencies between different words or tokens in a sentence. They have achieved state-of-the-art performance in various language-related tasks.

**NEED OF DEEP LEARNING**

The need for deep learning arises from the challenges posed by complex and abundant data in various domains. Here is a summary of the key reasons why deep learning is necessary:

1. **Complex Data Patterns:** Deep learning addresses the increasing complexity of data by leveraging deep neural networks that can automatically learn and extract meaningful representations from raw data. It allows for the capture of intricate patterns, relationships, and dependencies that are difficult to capture using traditional machine learning techniques.
2. **Feature Learning:** Deep learning eliminates the need for manual feature engineering by enabling models to learn hierarchical representations of data. This capability allows deep neural networks to automatically extract relevant features at different levels of abstraction, making them highly effective in handling diverse data types such as images, audio, text, and more.
3. **Non-linear Relationships:** Many real-world phenomena involve non-linear relationships and interactions. Deep learning models, with their non-linear activation functions, can capture and model complex non-linear patterns in the data, allowing for more accurate and nuanced predictions.
4. **End-to-End Learning:** Deep learning enables end-to-end learning, where models learn directly from raw input data to the desired output, without relying on manual feature extraction or intermediate representations. This approach eliminates the need for multiple stages of processing and improves overall performance by optimizing the entire data-to-output pipeline.
5. **Big Data and Scalability:** With the exponential growth of data, deep learning's scalability becomes essential. Deep neural networks can efficiently handle massive datasets and benefit from the availability of large-scale training data. This scalability allows deep learning models to learn from diverse and abundant data sources, leading to improved generalization and performance.
6. **Unstructured Data:** Deep learning is highly effective in handling unstructured data, such as images, videos, audio, and natural language. It can automatically learn complex representations and extract relevant features from unstructured data sources, making it ideal for tasks like image recognition, speech recognition, natural language processing, and more.

**Challenges in Deep Learning**

Deep learning can indeed be challenging, and there are certain scenarios where it may produce results that are not readily available through a simple Google search. Here are a few examples:

1. **Novel Image Generation:** Deep learning models such as Generative Adversarial Networks (GANs) can generate new and unique images that do not exist in reality. These models learn from existing data and can create visually appealing and original images that cannot be found on the internet.
2. **Creative Writing:** Deep learning models like language models can generate creative written content, including stories, poems, and even song lyrics. While these outputs can be inspired by existing works, they often present unique combinations and arrangements of words that are not directly searchable.
3. **Rare or Uncommon Topics:** Deep learning models can process vast amounts of text and learn patterns, but they may still struggle with rare or niche topics that have limited online presence. If the topic is not extensively covered on the internet, the results may be sparse or incomplete.
4. **Speculative or Hypothetical Scenarios:** Deep learning models can provide speculative answers based on the patterns and information they have learned, even if the specific information has not been explicitly provided or confirmed. In such cases, the results may not align with existing search results as they are based on inferred knowledge.
5. **Cutting-Edge Research:** Deep learning is a rapidly evolving field, and the latest research findings may not be immediately reflected in search engine results. New breakthroughs, algorithms, or architectures may take time to be documented, published, and disseminated, so upto-the-minute results may not be available through a simple search.

**Application of Deep Learning**

Deep learning has found numerous applications across various fields due to its ability to learn complex patterns and make accurate predictions. Here are some common applications of deep learning:

1. Computer Vision
2. Natural Language Processing
3. Speech Recognition and Synthesis
4. Recommendation System
5. Health Care
6. Autonomous Vehicles
7. Financial Services
8. Gaming and Robotics

**Types of Deep Learning**

There are several types of deep learning architectures and models, each designed to address specific tasks and data characteristics. Here are some commonly used types of deep learning:

* 1. **Feedforward Neural Networks:** Also known as Multi-Layer Perceptron’s (MLPs), these are the fundamental building blocks of deep learning. They consist of an input layer, one or more hidden layers, and an output layer. The information flows in one direction, from the input layer through the hidden layers to the output layer, without loops or feedback connections.
  2. **Convolutional Neural Networks (CNNs):** CNNs are widely used in computer vision tasks. They are designed to process data with a grid-like structure, such as images. CNNs leverage convolutional layers to automatically extract relevant features and hierarchical patterns from the input data. Pooling layers are used to reduce spatial dimensions, and fully connected layers perform the final classification or regression tasks.
  3. **Recurrent Neural Networks (RNNs):** RNNs are designed to process sequential and timeseries data, where the order and dependencies between elements matter. RNNs utilize recurrent connections that allow information to persist across time steps. This architecture enables them to capture temporal dependencies and context. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) are popular types of RNNs that address the vanishing gradient problem and improve information retention.
  4. **Generative Adversarial Networks (GANs):** GANs consist of two neural networks: a generator network and a discriminator network. GANs are used for generating new data that resembles the training data distribution. The generator network learns to create synthetic samples, while the discriminator network tries to distinguish between real and generated samples. The two networks compete against each other, leading to the generation of realistic and novel samples.
  5. **Autoencoders:** Autoencoders are unsupervised learning models that aim to learn efficient representations of the input data by reconstructing it from a compressed latent space. They consist of an encoder network that maps the input to a lower-dimensional representation and a decoder network that reconstructs the original input from the encoded representation. Autoencoders find applications in data compression, dimensionality reduction, and anomaly detection.
  6. **Transformer Networks:** Transformers have gained prominence in natural language processing tasks. They utilize self-attention mechanisms to capture relationships between different words or tokens in a sequence. Transformers excel in tasks such as machine translation, text summarization, and language generation. The popular model "BERT" (Bidirectional Encoder Representations from Transformers) is based on transformer architecture.

These are just a few examples of deep learning architectures. There are many other variations and specialized models designed for specific tasks or data types. Deep learning is a rapidly evolving field, and researchers are continually developing new architectures and models to tackle diverse challenges.

**Advantages of Deep Learning**

1. **Ability to Learn Complex Patterns**: Deep learning models possess the remarkable capability to learn and extract intricate patterns and representations from large volumes of data. By processing extensive datasets, these models automatically discover hierarchical features and relationships that may be challenging to define explicitly.
2. **High Accuracy**: Deep learning models have consistently demonstrated state-of-the-art performance across a wide range of tasks. Whether it's image classification, speech recognition, or natural language processing, these models can achieve remarkable accuracy levels, often surpassing traditional machine learning approaches.
3. **End-to-End Learning**: One of the notable advantages of deep learning is its ability to enable end-to-end learning. This means that models can learn directly from raw data without the need for manual feature engineering. By simplifying the development process, deep learning eliminates the requirement for domain-specific knowledge in certain cases.
4. **Adaptability to Large-Scale Data**: Deep learning excels at handling large-scale datasets efficiently. Models can process and learn from vast amounts of data, capitalizing on their capacity to generalize patterns and make accurate predictions. This adaptability to large-scale data is crucial in today's data-driven world.
5. **Flexibility and Versatility:** Deep learning models possess a remarkable level of flexibility and versatility. They can be applied to diverse tasks across different domains, including computer vision, natural language processing, speech recognition, and more. Furthermore, deep learning models can handle various types of data, such as images, text, audio, and video.
6. **Feature Extraction:** Deep learning models have the inherent capability to automatically learn relevant features from the input data. This eliminates the need for manual feature extraction, simplifying the modeling process and often leading to improved performance. By effectively extracting useful and discriminative features from raw data, deep learning models contribute to enhanced accuracy.
7. **Handling Unstructured Data:** Deep learning excels at processing unstructured data, such as images, audio, and text, which lack a fixed and predefined format. This capability makes deep learning particularly valuable in tasks that involve understanding and extracting information from unstructured sources. By effectively handling unstructured data, deep learning opens up new avenues for innovation.

**Disadvantages Of Deep Learning**

1. **Computational Resources:** Training deep learning models, especially those with large architectures, demands significant computational resources. Training neural networks with numerous layers and parameters can be computationally intensive and may require specialized hardware, such as powerful GPUs or even dedicated computing clusters. This high computational cost can pose a barrier to entry for individuals or organizations with limited resources.
2. **Overfitting:** Deep learning models are susceptible to overfitting, a phenomenon in which the model becomes too specialized in the training data and fails to generalize well to unseen data. Overfitting can occur when the model is overly complex relative to the available data or when the training dataset is insufficiently diverse. Proper regularization techniques, crossvalidation, and careful dataset management are necessary to mitigate overfitting.
3. **Interpretability and Explainability:** Deep learning models, particularly those with complex architectures like deep neural networks, are often considered black boxes. They lack interpretability and explainability, making it challenging to understand and interpret the underlying reasoning behind their predictions. This limitation can be problematic in domains where explainability and transparency are essential, such as healthcare or legal contexts.
4. **Vulnerability to Adversarial Attacks:** Deep learning models can be vulnerable to adversarial attacks, where small, imperceptible perturbations are added to input data with the goal of misleading the model's predictions. These attacks can have serious consequences, especially in critical applications such as autonomous vehicles or security systems. Developing robust models that are resistant to such attacks remains an active area of research.
5. **Data Dependency and Generalization:** Deep learning models heavily rely on the quality and representativeness of the training data. If the training dataset is biased, unrepresentative, or contains noisy or erroneous data, it can lead to biased or unreliable predictions. Additionally, deep learning models may struggle to generalize well to data outside the training distribution, resulting in reduced performance in real-world scenarios or on data that differs significantly from the training data.

**NATURAL LANGUAGE PROCESSING (NLP)**

Natural language processing (NLP) refers to the branch of computer science—and more specifically, the branch of [artificial intelligence or AI-co](https://www.ibm.com/topics/artificial-intelligence)ncerned with giving computers the ability to understand text and spoken words in much the same way human beings can.

NLP combines computational linguistics—rule-based modeling of human language—with statistical, machine learning, and deep learning models. Together, these technologies enable computers to process human language in the form of text or voice data and to ‘understand’ its full meaning, complete with the speaker or writer’s intent and sentiment.

NLP drives computer programs that translate text from one language to another, respond to spoken commands, and summarize large volumes of text rapidly—even in real time. There’s a good chance you’ve interacted with NLP in the form of voice-operated GPS systems, digital assistants, speech-to-text dictation software, customer service chatbots, and other consumer conveniences. But NLP also plays a growing role in enterprise solutions that help streamline business operations, increase employee productivity, and simplify mission-critical business processes.

**Python and the Natural Language Toolkit (NLTK)**

The Python programing language provides a wide range of tools and libraries for attacking specific NLP tasks. Many of these are found in the Natural Language Toolkit, or NLTK, an open source collection of libraries, programs, and education resources for building NLP programs.

The NLTK includes libraries for many of the NLP tasks listed above, plus libraries for subtasks, such as sentence parsing, word segmentation, stemming and lemmatization (methods of trimming words down to their roots), and tokenization (for breaking phrases, sentences, paragraphs and passages into tokens that help the computer better understand the text). It also includes libraries for implementing capabilities such as semantic reasoning, the ability to reach logical conclusions based on facts extracted from text.

**Statistical NLP, machine learning, and deep learning**

The earliest NLP applications were hand-coded, rules-based systems that could perform certain NLP tasks, but couldn't easily scale to accommodate a seemingly endless stream of exceptions or the increasing volumes of text and voice data.

Enter statistical NLP, which combines computer algorithms with machine learning and [deep learning mo](https://www.ibm.com/topics/deep-learning)dels to automatically extract, classify, and label elements of text and voice data and then assign a statistical likelihood to each possible meaning of those elements. Today, deep learning models and learning techniques based on convolutional neural networks (CNNs) and recurrent neural networks (RNNs) enable NLP systems that 'learn' as they work and extract ever more accurate meaning from huge volumes of raw, unstructured, and unlabeled text and voice data sets.

**4. ARCHITECTURE**

###### ARCHITECTURE

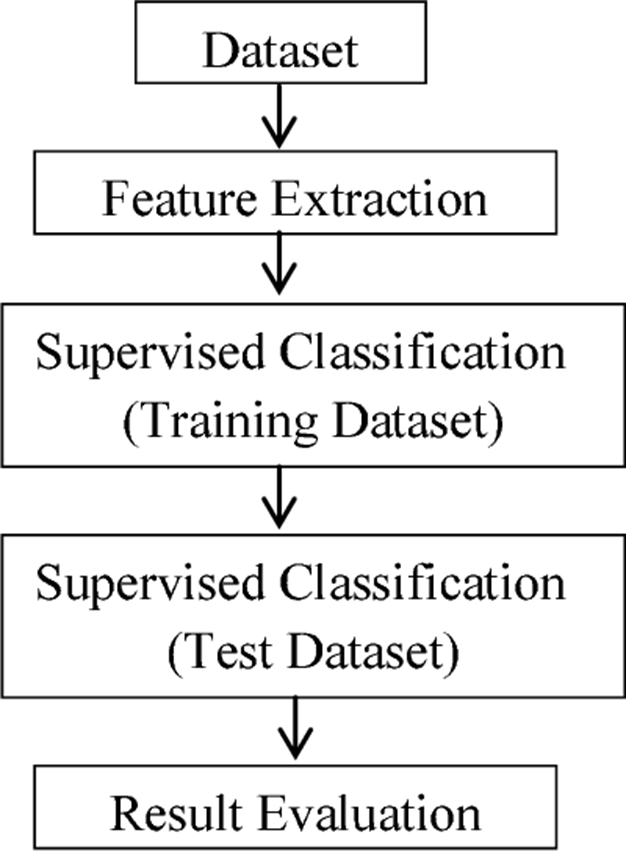
****

Figure 4.1: The work flow of the project

4.1 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

**GOALS:**

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

4.1.1 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

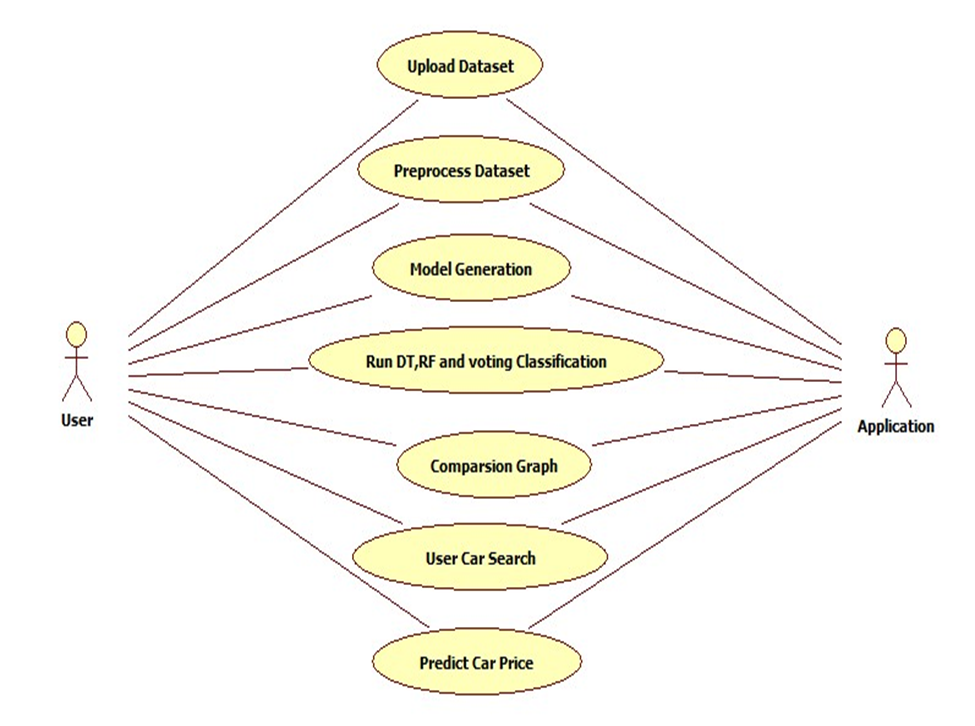


Figure 4.2 : Use case Diagram for car price prediction using machine learning

4.1.2 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

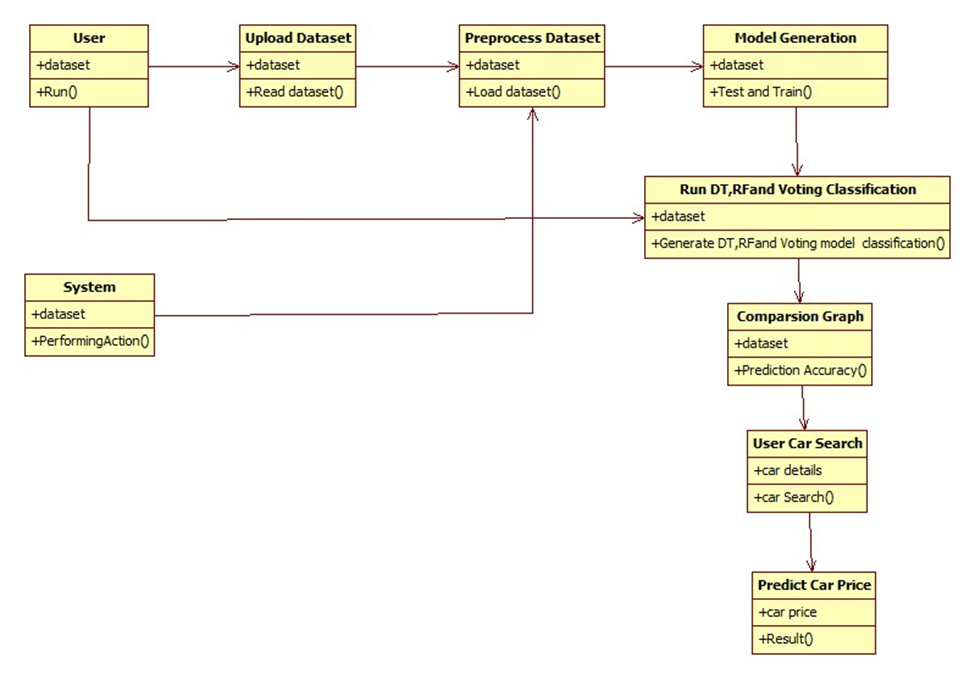


Figure 4.3: Class Diagram for system, car price prediction.

4.1.3 SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

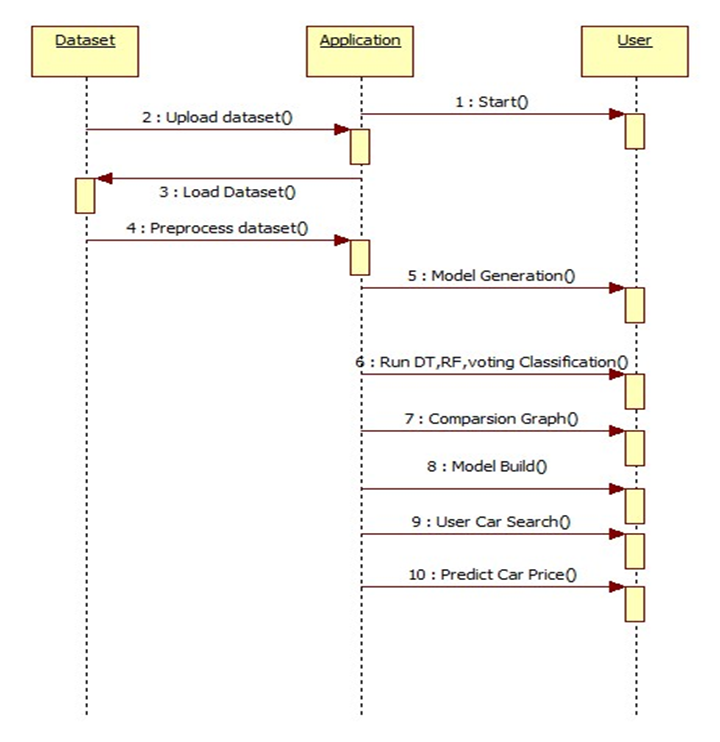


Figure 4.4: Sequence Diagram for car price prediction.

**4.1.4 ACTIVITY DIAGRAM**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

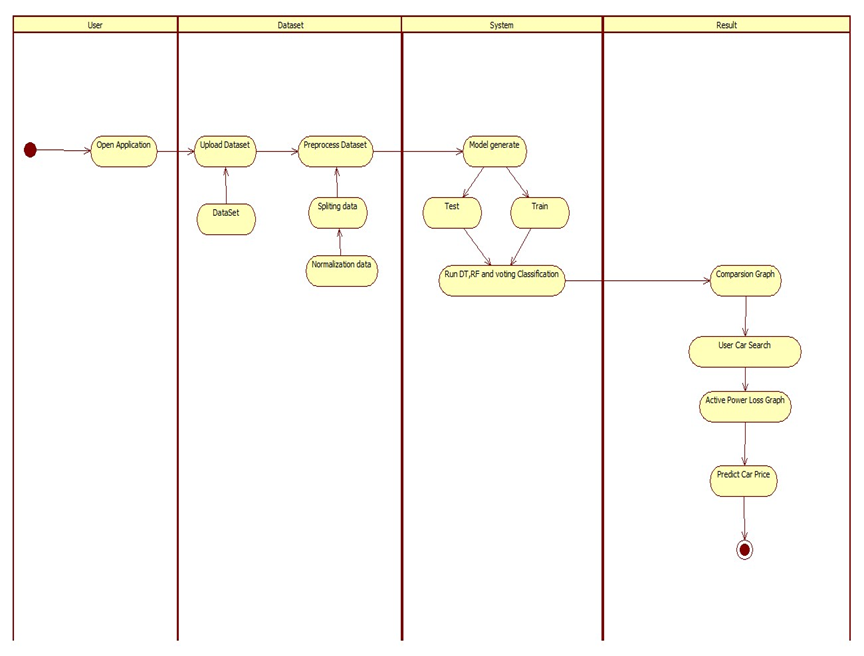


Figure3.3: Activity Diagram for car price prediction.

**5. IMPLENENTATION**

###### 5. IMPLEMENTATION

5.1 FRONT-END(WEBAPP.PY)

**1. home.html**

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-tofit=no">

<link rel="stylesheet" href="static/css/style.css">

<!-- Bootstrap CSS -->

<linkrel="stylesheet"

href="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/css/bootstrap.min.css"

integrity="sha384-

MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPM

O" crossorigin="anonymous">

<title>Car Price Predictor</title>

</head>

<body class="backgroundColor">

<div class=" d-flex flex-column flex-md-row align-items-center p-3 px-md-4 mb-3

navbar-light" style="background-color: #e0f2f1;">

<h5 class="my-0 mr-md-auto font-weight-normal"><b><h4>CAR PRICE

PREDICTOR</h4></b></h5>

<nav class="my-2 my-md-0 mr-md-3 ">

<a class="p-2 text-dark" href="{{url\_for('home')}}"><b>Home</b></a>

</nav>

<a class="btn btn-outline-primary" href="/logout">Log out</a>

</div>

<div class="container">

<div clas="row">

<div class="card mt-50" style="width:100%;height:100%">

<div class="card-header">

<div class="col-12" style="text-align:center">

<</div>

</div>

<div class="card-body">

<form class="form" method="post" >

<div class="col-10 form-group" style="text-align: center">

<label> <b>Select company: </b></label>

<select class="selectpicker form-control" id="company" name="company"

required="1" onchange="load\_car\_models(this.id,'car\_model')">

{% for company in companies %}

<option value="{{company}}">{{company}} </option>

{% endfor %}

</select>

</div>

<div class="col-10 form-group" style="text-align: center">

<label> <b>Select Model: </b></label>

<select class="selectpicker form-control" id="car\_model" name="car\_model"

required="1">

</select>

</div>

<div class="col-10 form-group" style="text-align: center">

<label> <b>Select Year of Purchase: </b></label>

<select class="selectpicker form-control" id="year" name="year" required="1">

{% for year in years %}

<option value="{{year}}">{{year}} </option>

{% endfor %}

</select>

</div>

<div class="col-10 form-group" style="text-align: center">

<label> <b>Select Fuel Type: </b></label>

<select class="selectpicker form-control" id="fuel\_type" name="fuel\_type"

required="1">

Welcome to Car Price Predictor</h1>

{% for fuel\_type in fuel\_types %}

<option value="{{fuel\_type}}">{{fuel\_type}} </option>

{% endfor %}

</select>

</div>

<div class="col-10 form-group" style="text-align: center">

<label> <b>Kilometers travelled: </b></label>

<input class="form-control" type="text" id="kms\_driven" name="kms\_driven"

placeholder="Enter no.of kms travelled" >

</input>

</div>

<div class="col-10 form-group" style="text-align: center">

<button class="btn btn-primary btn-block btn-lg" onclick="send\_data()"

value="Predict">Predict Price</button>

</div>

</form>

<br>

<div class="row">

<div class="col-12" style="text-align: center">

<h3><span id="prediction"></span> </h3>

</div>

</div>

</div>

</div>

</div>

</div>

<script>

function load\_car\_models(company\_id,car\_model\_id)

{

var company= document.getElementById(company\_id);

var car\_model= document.getElementById(car\_model\_id);

car\_model.value="";

car\_model.innerHTML="";

{% for company in companies %}

if(company.value == "{{company}}" )

{

{% for model in car\_models %}

{% if company in model %}

var newOption = document.createElement("option");

newOption.value="{{ model }}";

newOption.innerHTML="{{ model }}";

car\_model.options.add(newOption);

{% endif %}

{% endfor %}

}

{% endfor %}

}

function form\_handler()

{

event.preventDefault();

}

function send\_data()

{

document.querySelector('form').addEventListener('submit', form\_handler);

var fd= new FormData(document.querySelector('form'));

var xhr=new XMLHttpRequest();

xhr.open('POST', '/predict', true);

document.getElementById("prediction").innerHTML="wait! predicting price...";

xhr.onreadystatechange= function()

{

if(xhr.readyState == XMLHttpRequest.DONE)

{

document.getElementById("prediction").innerHTML="The Predicted Price is: "+

xhr.responseText + " Rs/-";

}

}

xhr.onload=function(){};

xhr.send(fd);

}

</script>

<!-- Optional JavaScript -->

<!-- jQuery first, then Popper.js, then Bootstrap JS -->

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-

q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"

crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.3/dist/umd/popper.min.js"

integrity="sha384-

ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49"

crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/js/bootstrap.min.js"

integrity="sha384-

ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"

crossorigin="anonymous"></script>

</body>

</html>

5.2 BACK-END(DETECT.PY)

**1. App.java**

import pandas as pd

#from flask import Flask, render\_template, request, url\_for,redirect,session

import pickle

import numpy as np

from flask import \*

import flask\_login

import os

from num2words import num2words

import mysql.connector

model=pickle.load(open("LinearRegressionModel.pkl",'rb'))

car=pd.read\_csv("cleaned car.csv")

app=Flask(\_\_name\_\_)

app.secret\_key=os.urandom(24)

conn=mysql.connector.connect(

host='localhost',

user='root',

password='Password123@',

port='3306',

database='database'

)

mycursor=conn.cursor()

@app.route('/')

def login():

if 'user\_id' in session:

return redirect('/home')

else:

return render\_template('login.html')

@app.route('/register')

def register():

return render\_template('register.html')

@app.route('/logout')

def logout():

session.pop('user\_id')

return redirect('/')

@app.route('/login\_validation',methods=['POST'])

def login\_validation():

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

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

mycursor.execute('''SELECT \* FROM `database`.`uinfo` WHERE `email` LIKE '{}'

AND `pwd` LIKE '{}' '''

.format(email,password))

uinfo=mycursor.fetchall()

if len(uinfo)>0:

session['user\_id']=uinfo[0][0]

return redirect('/home')

else:

flash('Incorrect username/ password')

return redirect('/')

@app.route('/add\_user',methods=['POST'])

def add\_user():

name=request.form.get('uname')

email=request.form.get('uemail')

password=request.form.get('upassword')

mycursor.execute('''INSERT INTO `database`.`uinfo` (`uid`,`name`, `email`, `pwd`)

VALUES (NULL , '{}','{}','{}' )'''.format(name, email, password))

conn.commit()

return render\_template('login.html')

@app.route('/home')

def home():

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")

year.insert(0,"Select Year of Purchase")

if 'user\_id' in session:

return

render\_template('home.html',companies=companies,car\_models=car\_models,years=year,

fuel\_types=fuel\_type)

else:

return redirect('/')

@app.route('/predict',methods=['POST'])

def predict():

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

car\_model=request.form.get('car\_model')

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

fuel\_type=request.form.get('fuel\_type')

kms\_driven=request.form.get('kms\_driven')

prediction=model.predict(pd.DataFrame([[car\_model, company, year, kms\_driven,

fuel\_type]], columns=['name', 'company', 'year', 'kms\_driven', 'fuel\_type']))

return str(np.round(prediction[0], 0))

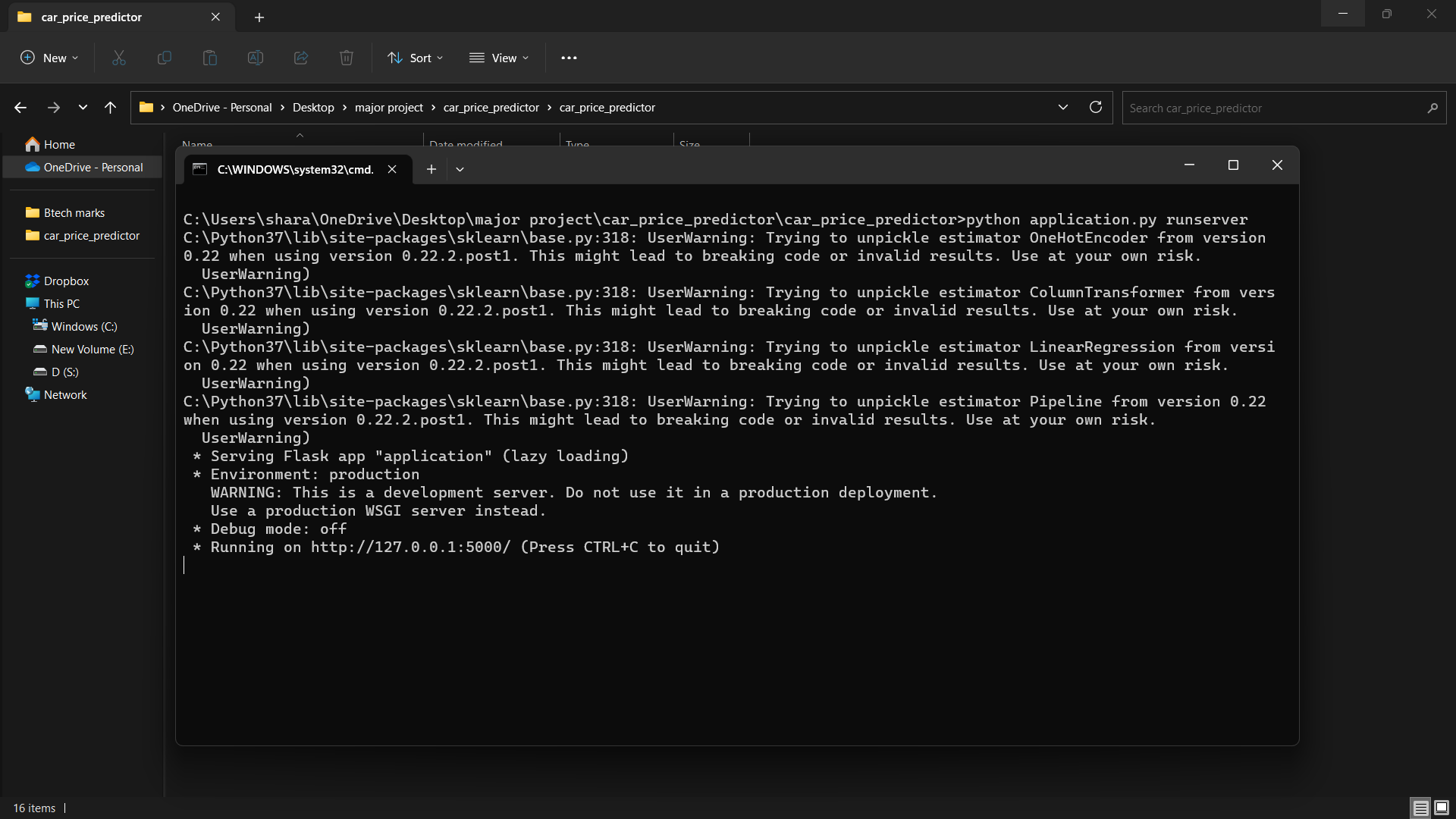
if \_\_name\_\_=="\_\_main\_\_":

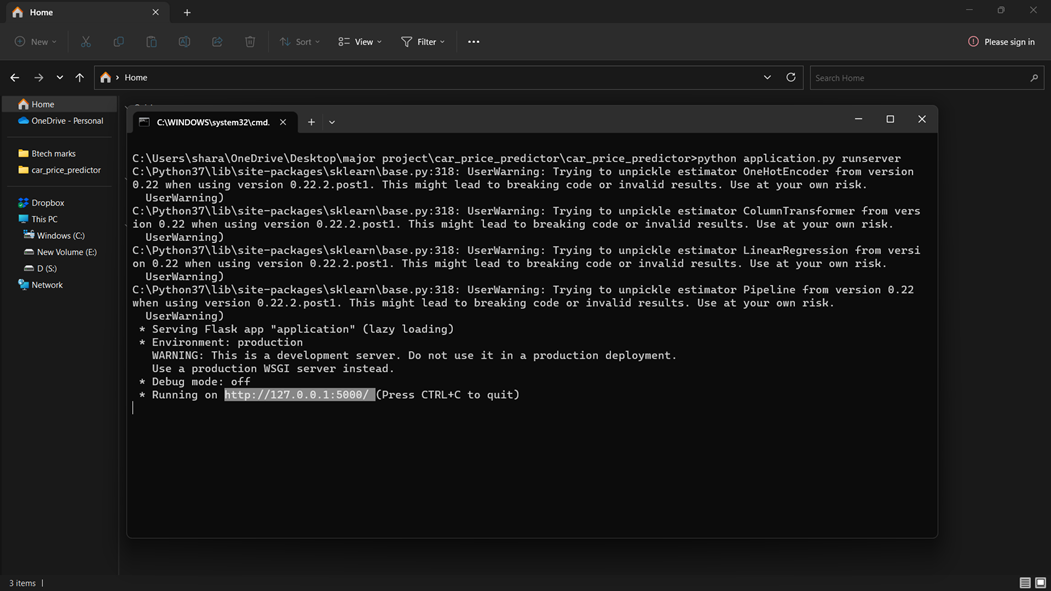
app.run(debug=True)

**6. SCREENSHOTS**

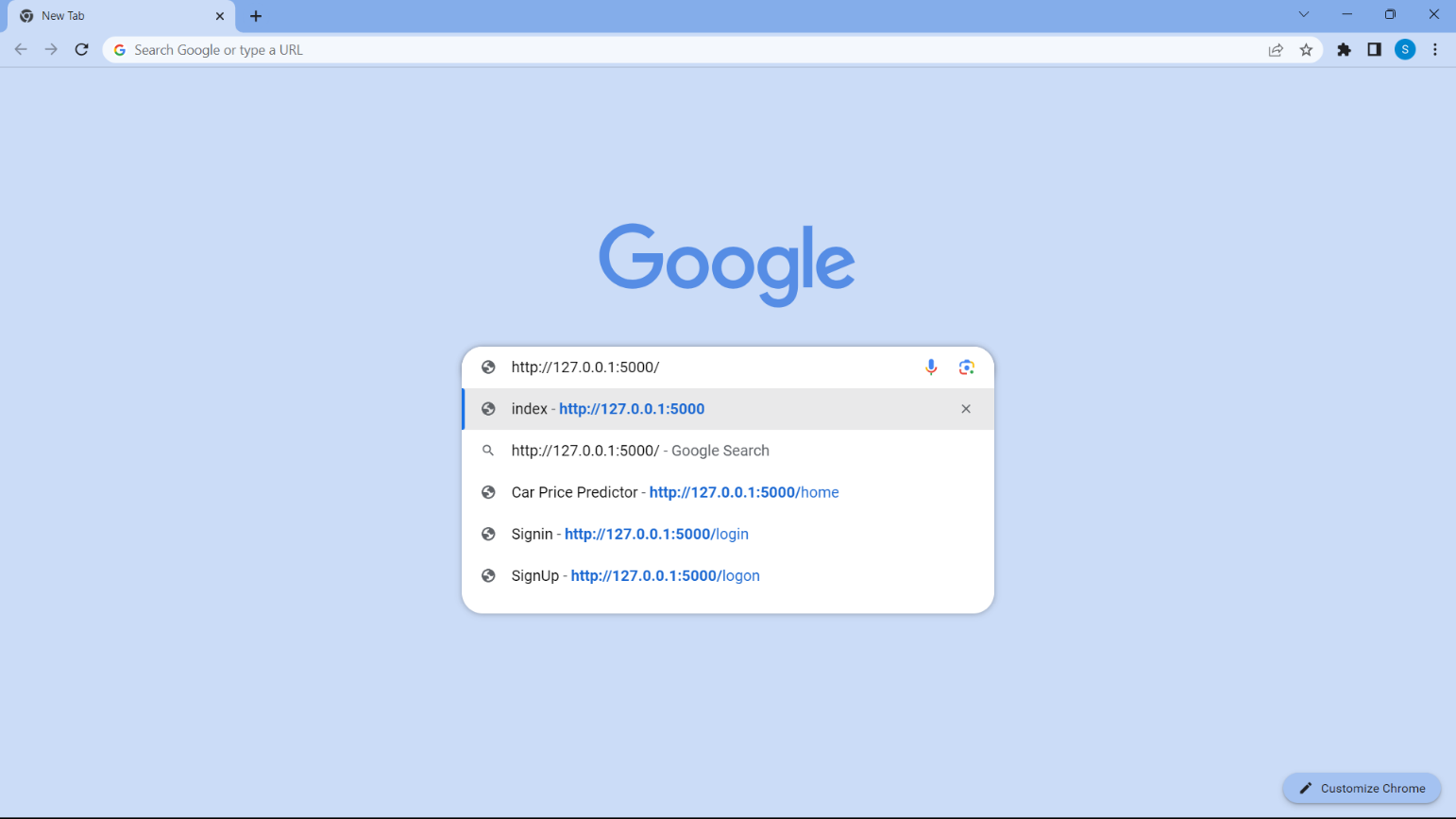
###### 6. SCREENSHOTS

###### 6.1 LAYOUT OF TESTING PLATFROM

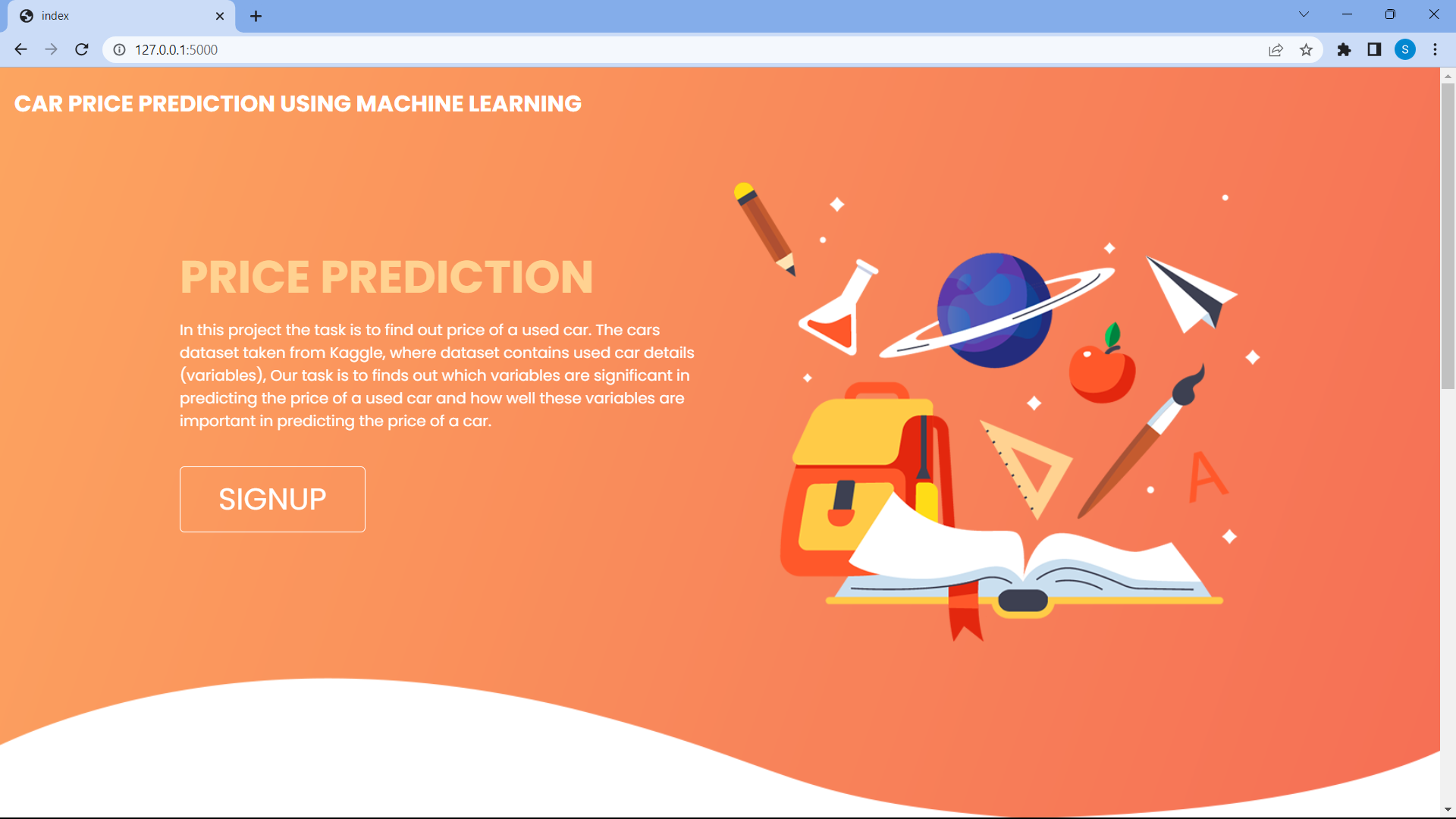




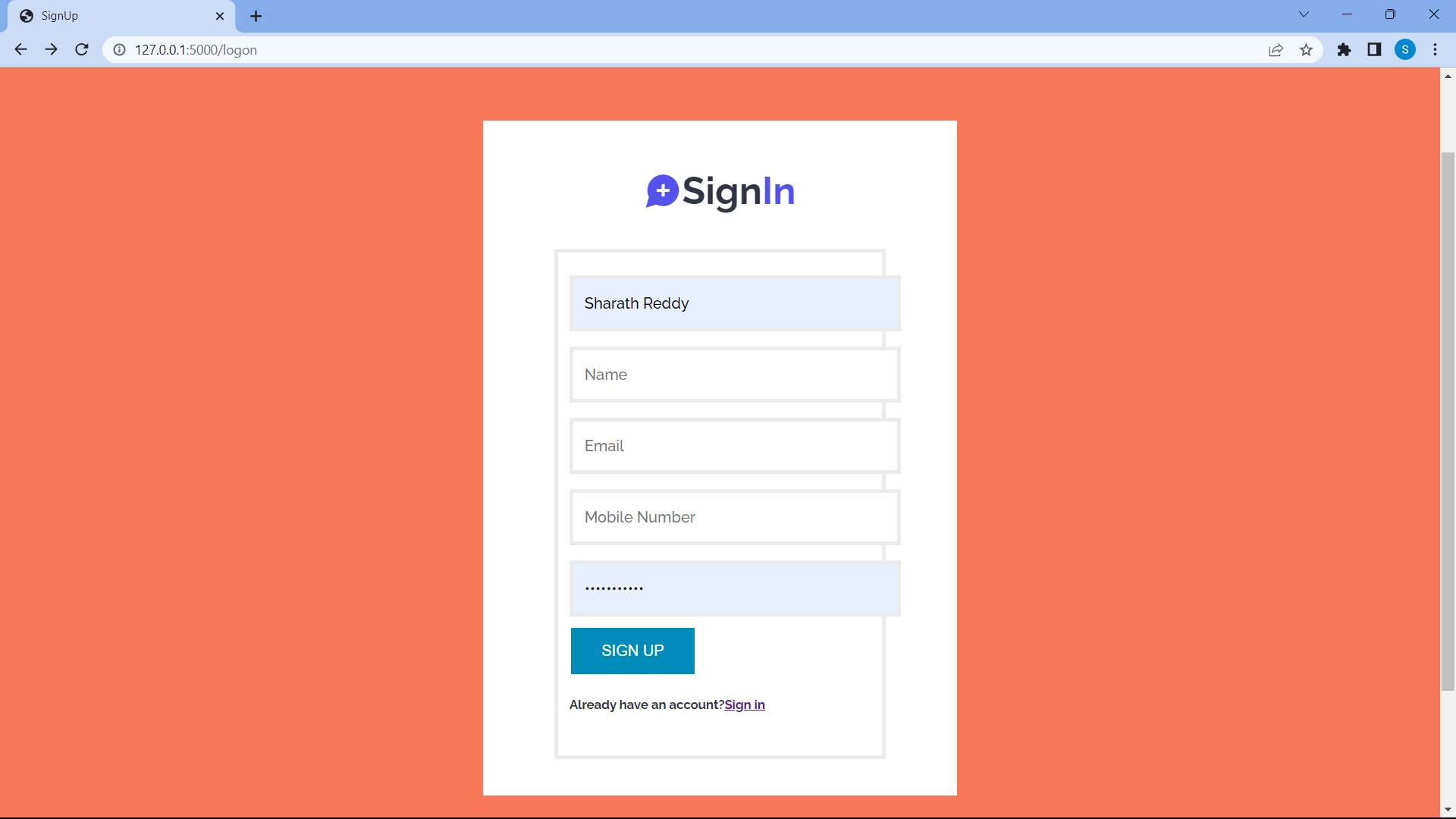
**6.2 GO TO THE GOOGLE AND PAST THE HOST LOCATION**



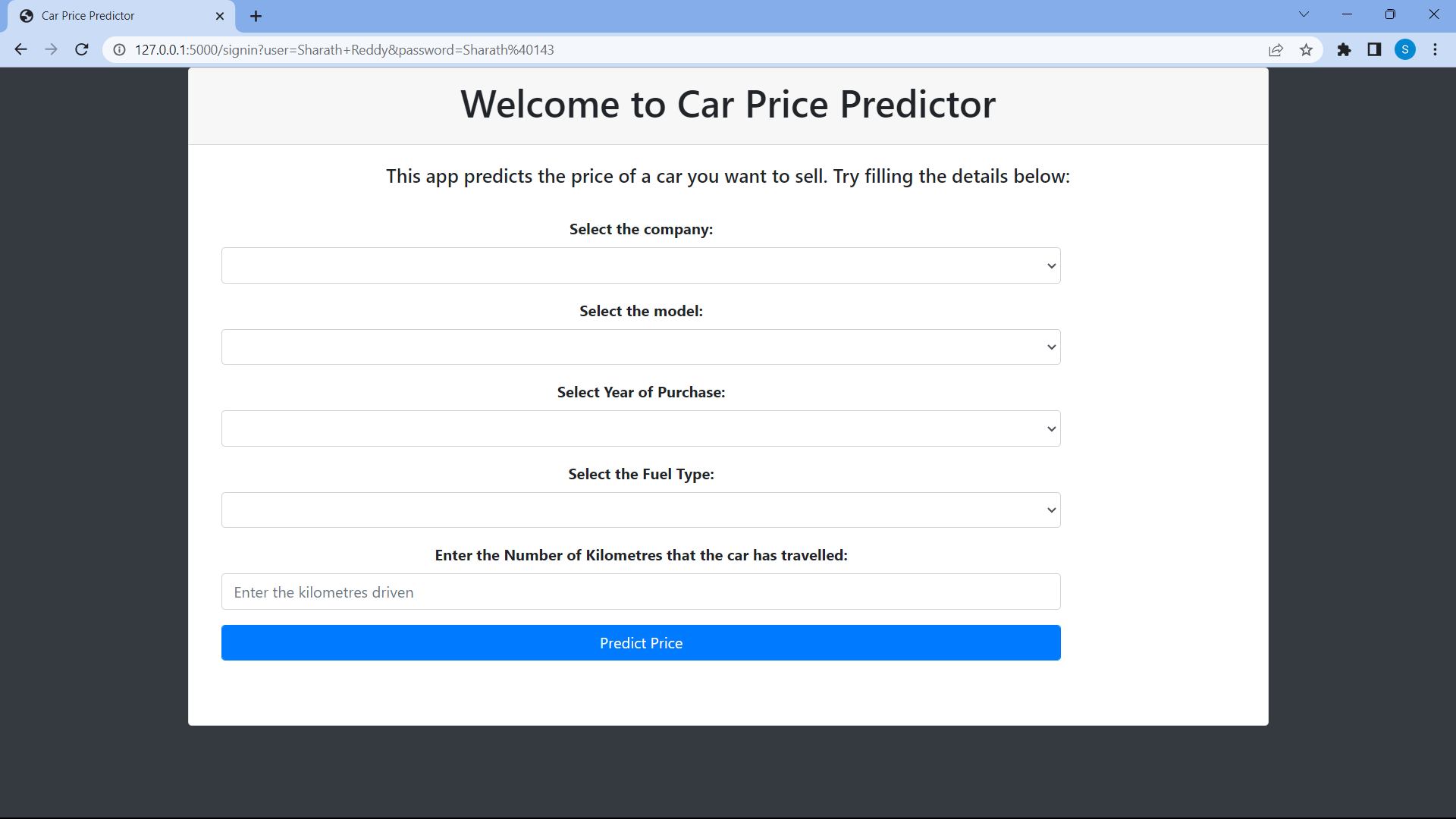
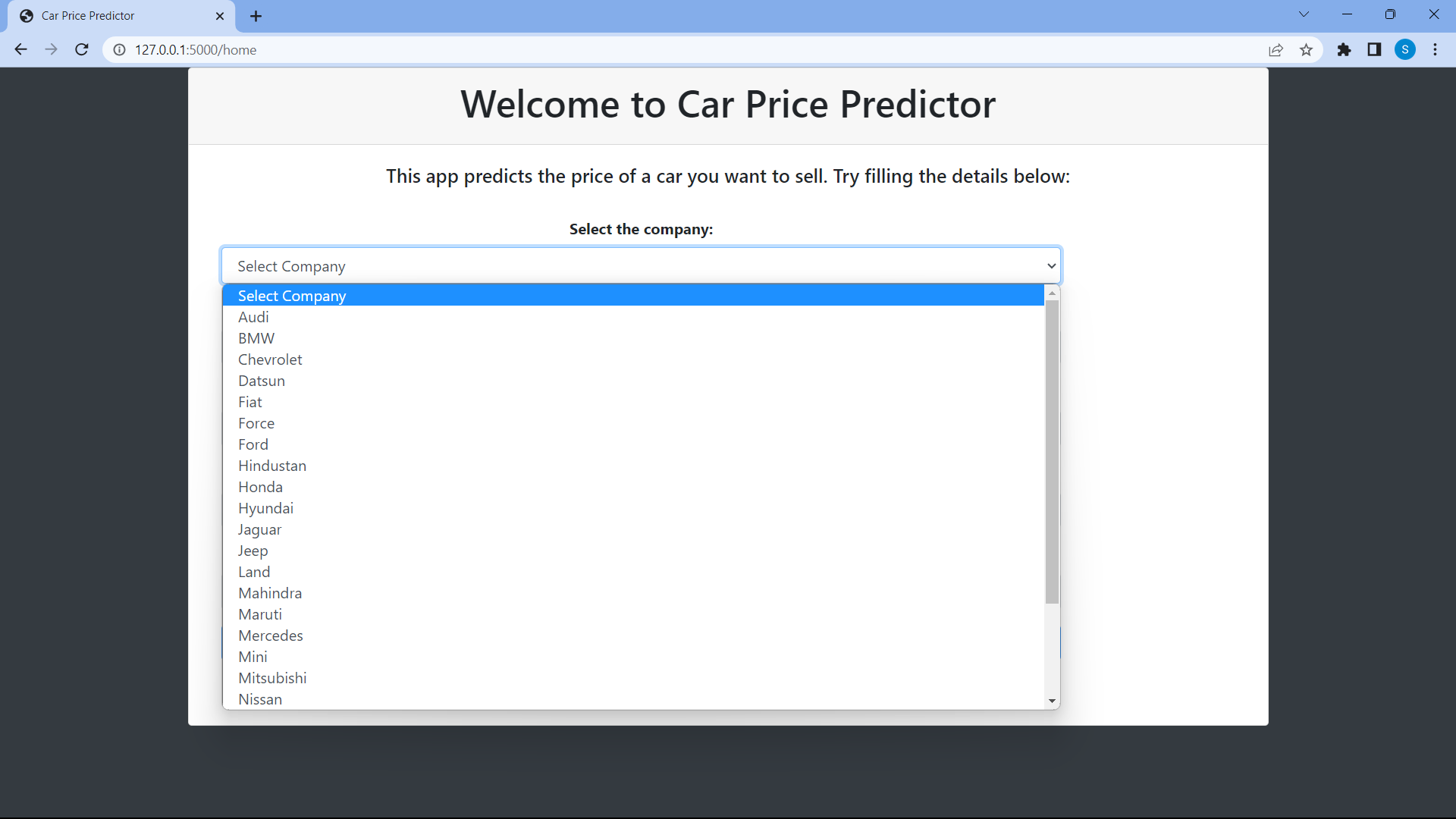
**6.3 THIS THE HOME WEB PAGE**

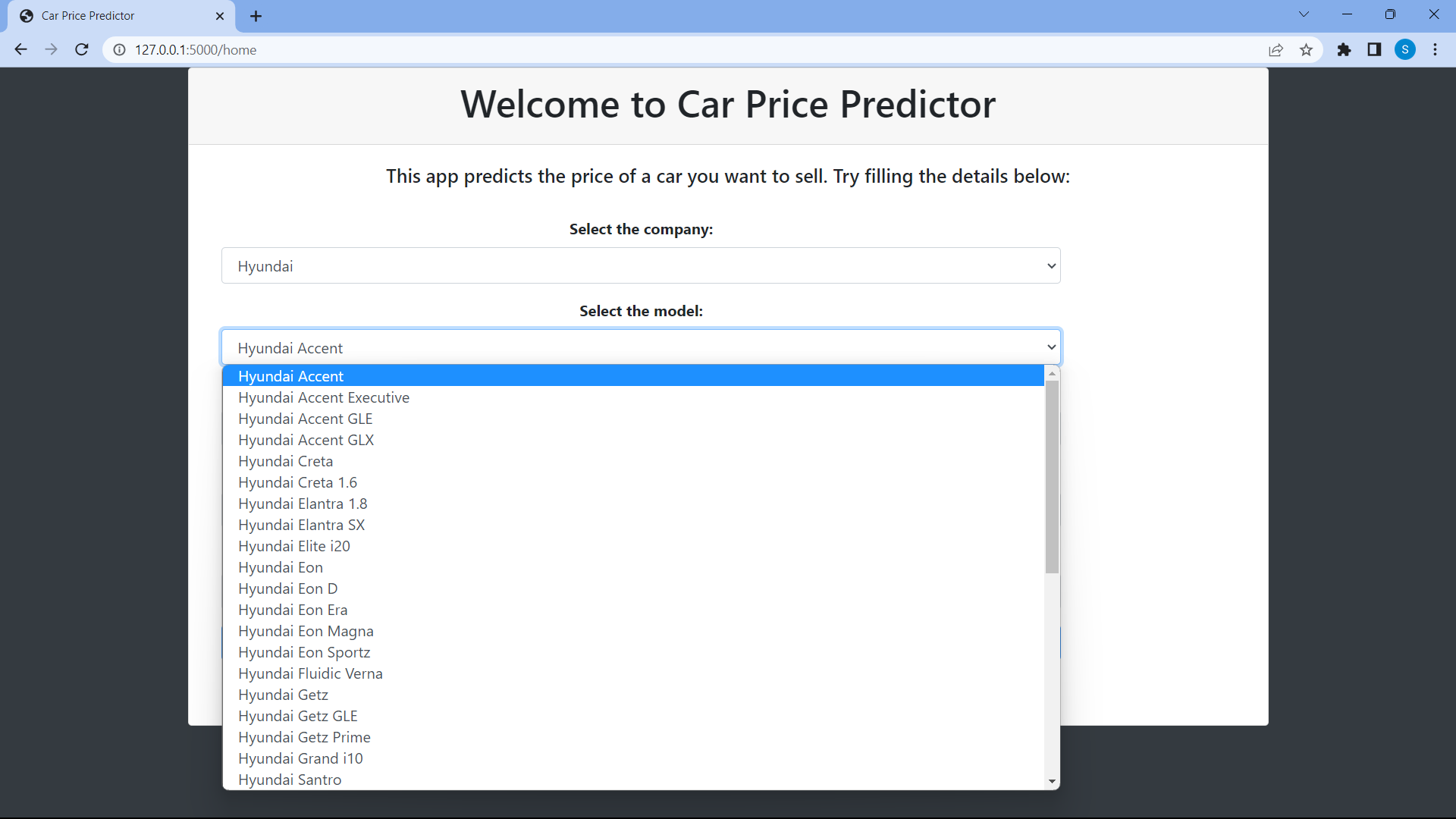
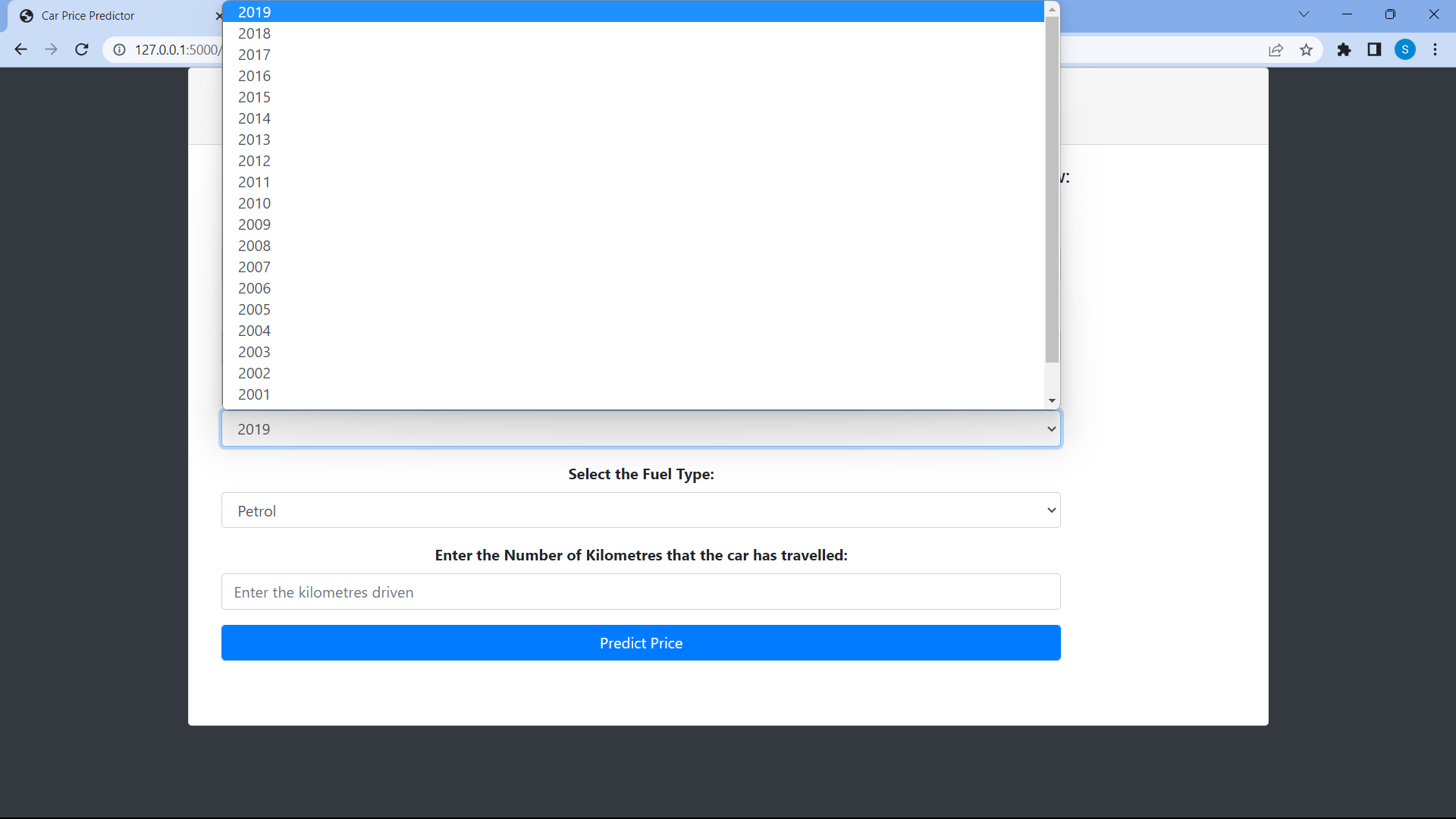


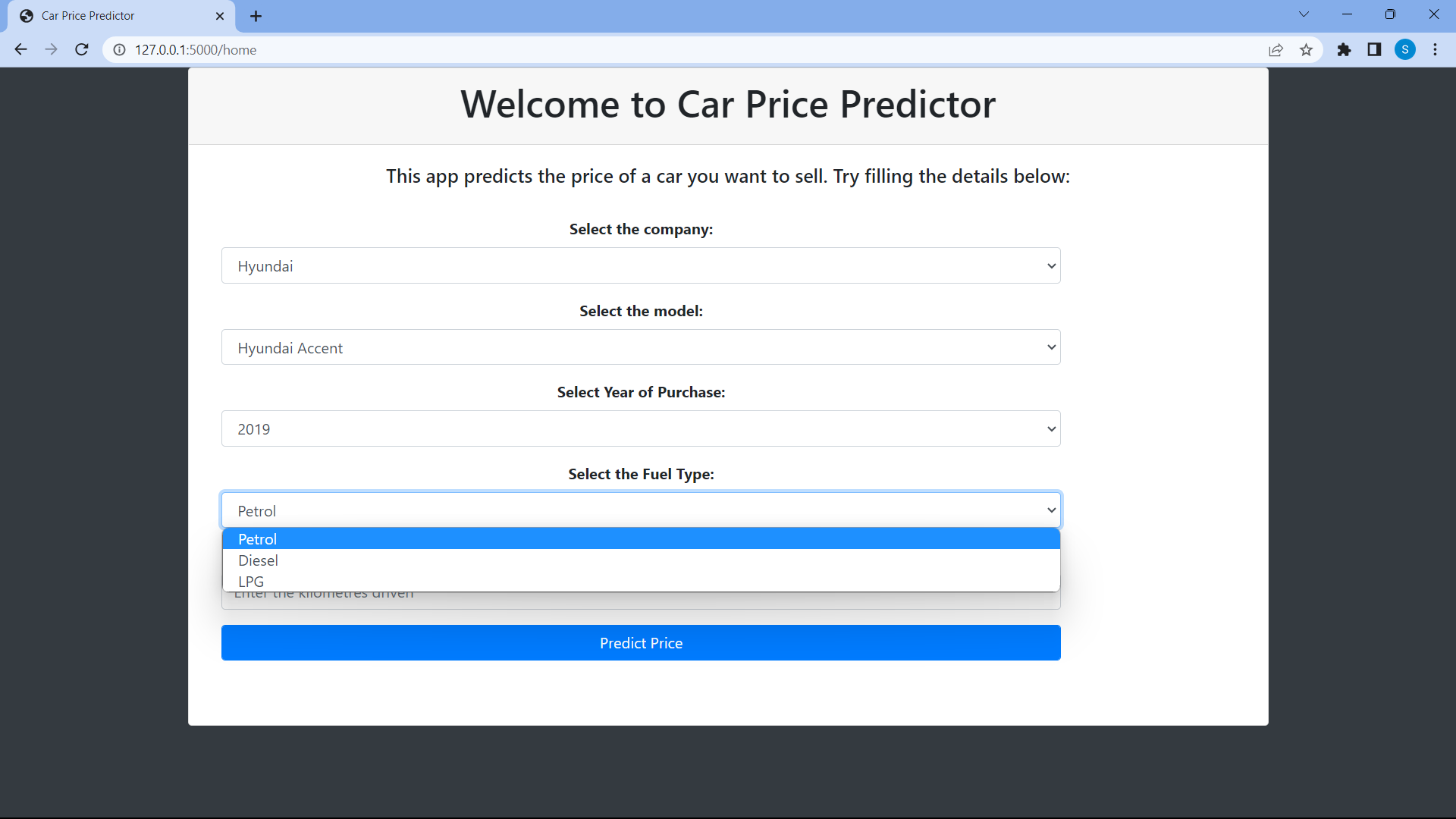
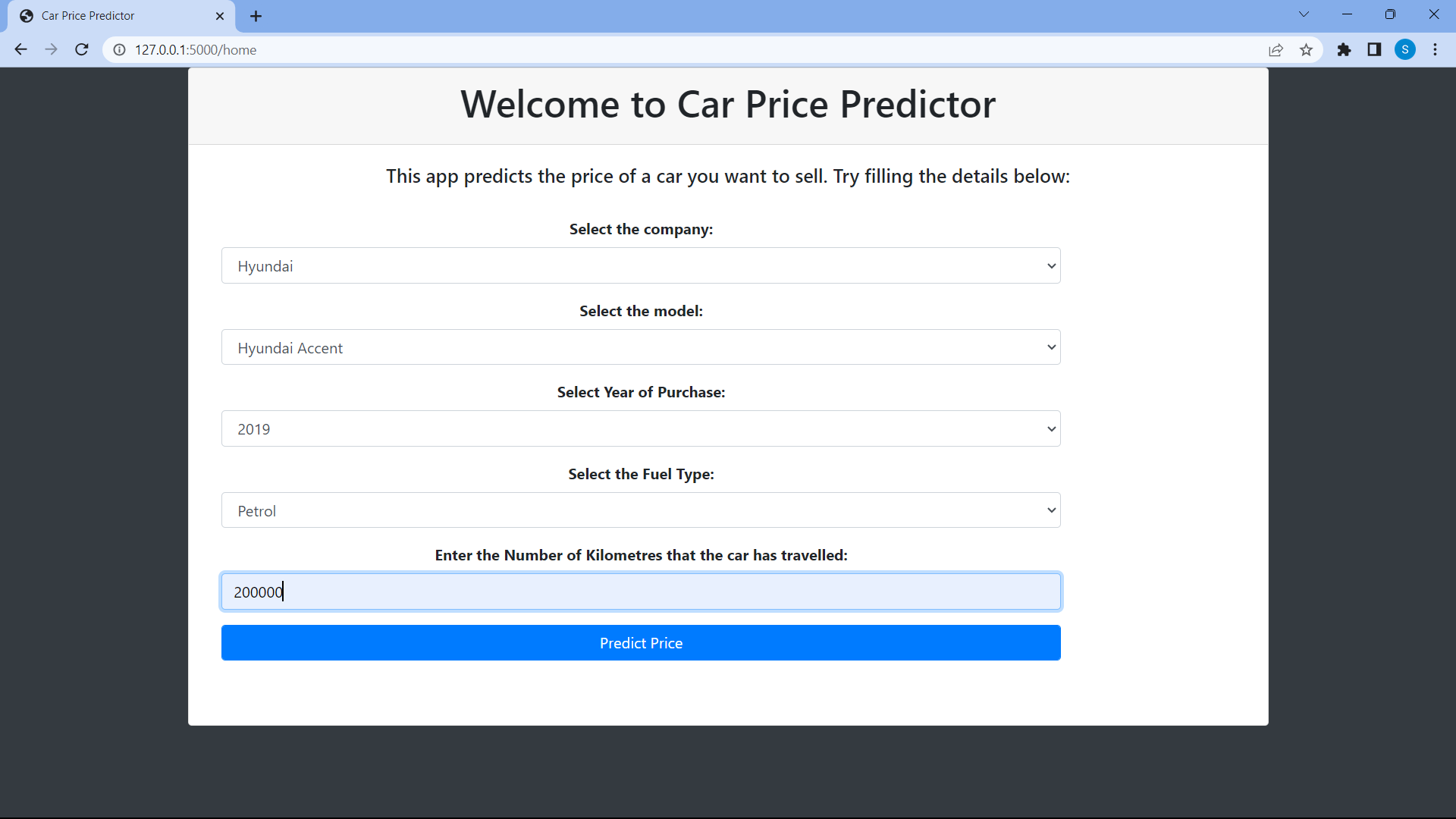
**6.4 REGISTER PAGE OF WEB APPLICATION**



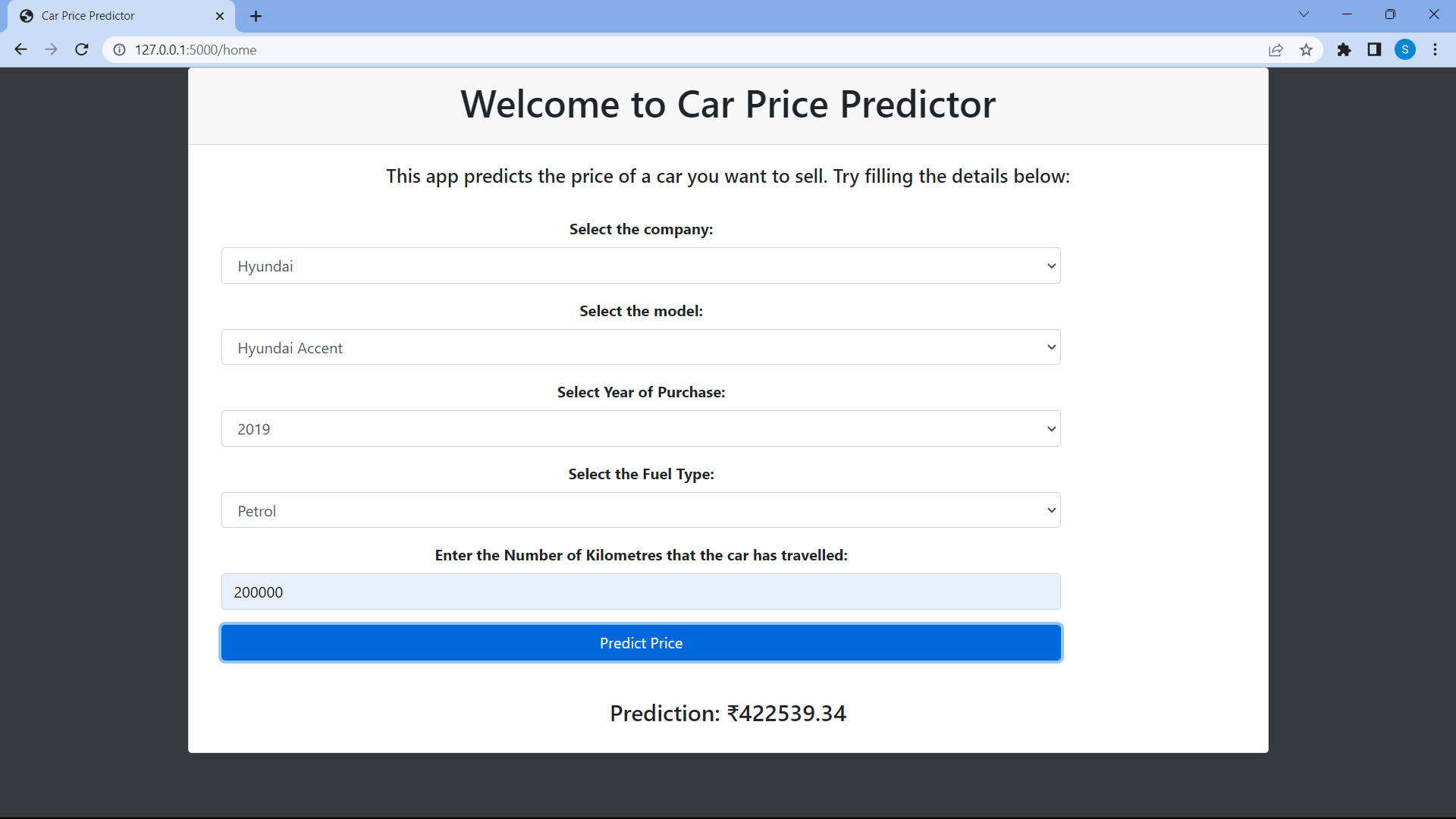
**6.5 LOGIN PAGE OF WEB APPLICATION **

**6.6 HOME PAGE OF WEB APPLICATION6.7 DISPLAYING AVAILABLE CAR COMPANIES**

**6.8 DISPLAYING SUITABLE CAR MODEL6.9 DISPLAYING AVAILABLE YEARS**

**6.10 DISPALYING AVAILABLE FUEL TYPE6.11 ENTER THE KILOMETERS TRAVELLED**

**6.12 DISPLAYING PREDICTED PRICE**

****

**7. TESTING**

###### 7. TESTING

7.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

7.2 TYPES OF TESTING

7.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

7.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centred on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Unit Testing**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.Test strategy and approach Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

• All field entries must work properly.

• Pages must be activated from the identified link.

• The entry screen, messages and responses must not be delayed.

**Features to be tested**

• Verify that the entries are of the correct format

• No duplicate entries should be allowed

• All links should take the user to the correct page.

**Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results**: All the test cases mentioned above passed successfully. No defects encountered.

7.3 TEST CASES

**USER REQUIREMENTS:**

1. **Home**

|  |  |
| --- | --- |
| **Use case ID** | **PREDICTING THE PRICE OF USED CARS USING MACHINE LEARNING TECHNIQUES** |
| Use case Name | Home button |
| Description | Display home page of application |
| Primary actor | User |
| Precondition | User must open application |
| Post condition | Display the Home Page of an application |
| Frequency of Use case | Many times |
| Alternative use case | N/A |
| Use case Diagrams |  |
| Attachments | N/A |

**8. CONCLUSION**

###### 8.1 CONCLUSION

Car price prediction can be a challenging task due to the high number of attributes that should be considered for the accurate prediction. The major step in the prediction process is collection and preprocessing of the data. In this paper, we check out the software of supervised device learning strategies to are expecting the charge of used automobiles. Different strategies (Regression Algorithms) are used to make the predictions. The predictions are then evaluated and in comparison on the way to discover the ones which offer the best performances. Also, the post-hoc test revealed that the error rates in multiple regression models and lasso regression models aren’t significantly different from each other. To get even more accurate models, we can also choose more advanced machine learning algorithms such as random forests, an ensemble learning algorithm which creates multiple decision/regression trees, which brings down overfitting massively or Boosting, which tries to bias the overall model by weighing in the favor of good performers. More data from newer websites and different countries can also be scraped and this data can be used to retrain these models to check for reproducibility.

###### 8.2 FUTURE SCOPE

The main limitation of this study is the low number of records that have been used.As future work, we intend to collect more data and to use more advanced techniques like artificial neural networks, fuzzy logic and genetic algorithms to predict car prices.

**9. BIBLIOGRAPHY**

###### 9. BIBLIOGRAPHY

**9.1 GITHUB LINK**

Github link: https://github.com/discover2722/majorproject

9.2 REFERENCES

[1] NATIONAL TRANSPORT AUTHORITY. 2014. Available from: http://nta.gov.mu/English/Statistics/Pages/Archive.aspx [Accessed 15 January 2014].

[2] MOTORS MEGA. 2014. Available from: http://motors.mega.mu/news/2013/12/17/auto-market-8-decrease-sales-newcars/ [Accessed 17 January 2014].

[3] LISTIANI, M., 2009. Support Vector Regression Analysis for Price Prediction in a Car Leasing Application. Thesis (MSc). Hamburg University of Technology.

[4] RICHARDSON, M., 2009. Determinants of Used Car Resale Value. Thesis (BSc). The Colorado College.

[5] WU, J. D., HSU, C. C. AND CHEN, H. C., 2009. An expert system of price forecasting for used cars using adaptive neuro-fuzzy inference. Expert Systems with Applications. Vol. 36, Issue 4, pp. 7809-7817.

[6]. S. Pudaruth, ―Predicting the price of used cars using machine learning techniques,‖ Int. J. Inf. Comput. Technol,vol. 4, no. 7, pp. 753–764, 2014. 183 Authorized licensed use limited to: Carleton University. Downloaded on May 29,2021 at 09:56:13 UTC from IEEE Xplore. Restrictions apply.

[7]. M. Jayakameswaraiah and S. Ramakrishna, ―Development of data mining system to analyze cars using tknn clustering algorithm,‖ International Journal of Advanced Research in Computer Engineering Technology, vol.3, no. 7, 2014.

[8]. Q. Yuan, Y. Liu, G. Peng, and B. Lv, ―A prediction study on the car sales based on web search data,‖ in The International Conference on E-Business and E-Government (Index by EI), 2011, p. 5.

[9]. K. S. Durgesh and B. Lekha, ―Data classification using support vector machine,‖ Journal of theoretical and applied information technology, vol. 12, no. 1, pp. 1–7, 2010.

[10]. Coronavirus on used car saleshttps://economictimes.indiatimes.com/topic/coronavirus-impact-onused-car-sales.

[11]. no. 22, pp. 12 693–12 700, 2018. [12] E. Gegic, B. Isakovic, D. Keco, Z. Masetic, and J. Kevric, ―Car price prediction using machine learning techniques,‖ 2019.

[12]. N. Pal, P. Arora, P. Kohli, D. Sundararaman, and S. S. Palakurthy, ―How much is my car worth? a methodology for predicting used cars prices using random forest,‖ in Future of Information and Communication Conference. Springer, 2018, pp. 413–422.

[13]. R. Ragupathy and L. Phaneendra Maguluri, ―Comparative analysis of machine learning algorithms on social media test,‖ International Journal of Engineering and Technology(UAE), vol. 7, pp. 284–290, 03 2018.

[14]. F. Harahap, A. Y. N. Harahap, E. Ekadiansyah, R. N. Sari, R. Adawiyah, and C. B. Harahap, ―Implementation of naive Bayes classification method for predicting purchase,‖ in 2018 6th International Conference on Cyber and IT Service Management (CITSM). IEEE, 2018, pp. 1–5.

[15]. F. Osisanwo, J. Akinsola, O. Awodele, J. Hinmikaiye, O. Olakanmi, and J. Akinjobi,-Supervised machine learning algorithms: classification and comparison,‖ International Journal of Computer Trends and Technology (IJCTT), vol. 48, no. 3, pp. 128–138, 2017.