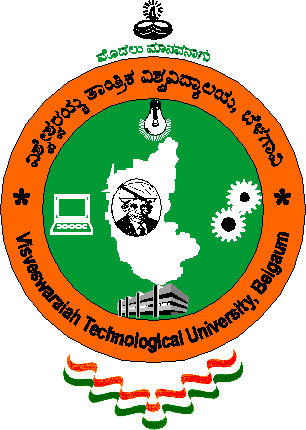
# VISVESVARAYA TECHNOLOGICAL UNIVERSITY BELGAUM-590014

****

**A Mini-Project Report On**

***“CAR DEALERSHIP DATABASE MANAGEMENT SYSTEM”***

*A Mini-project report submitted in partial fulfillment of the requirements for the award of the degree of* ***Bachelor of Engineering in Computer Science and Engineering*** *of*

*Visvesvaraya Technological University, Belgaum.*

Submitted by:

**RAHUL MISHRA [1AM19CS159]**

**RAHUL GOOGIKOLL [1AM19CS158]**

**SAURAV KUMAR [1AM19CS195]**

**OM PRAKASH JHA [1AM19CS133]**

**D KARTHIK REDDY [1AM20CS401]**

Under the Guidance of:

Veena Bhat

**(Asst. Prof., Dept. of CSE)**



**Department of Computer Science and Engineering**

## AMC Engineering College,

#### 18th K.M, Bannerghatta Main Road, Bangalore-560 083

2021-2022

**AMC Engineering College,**

18th K.M, Bannerghatta Main Road, Bangalore-560 083

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



## CERTIFICATE

This is to certify that the mini-project work entitled **“HOTEL MANAGEMENT DATABASE SYSTEM”** has been successfully carried out by **RAHUL MISHRA (1AM19CS159), RAHUL GOOGIKOLL (1AM19CS158), SAURAV KUMAR**

**(1AM19CS195) and OM PRAKASH JHA (1AM19CS133),** bonafide students of **AMC Engineering College** in partial fulfillment of the requirements for the award of degree in **Bachelor of Engineering in Computer Science and Engineering** of **Visvesvaraya Technological University, Belgaum** during academic year 2021-2022. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of project work for the said degree.

**Guide:**

|  |  |  |
| --- | --- | --- |
| **Mrs. Veena Bhat** | **Dr . K Ananthapadmanabha** | **Dr. Rajesh S** |
| **Asst. Prof., Dept. of CSE** | **HOD, Dept. of CSE** | **Principal, AMCEC** |
| **Examiners: 1.** |  | **Signature with Date** |
| **2.** |  |  |

# ACKNOWLEDGEMENT

It gives us immense pleasure to present before you our project titled **‘CAR DEALERSHIP DATABASE MANAGEMENT SYSTEM’.** The joy and satisfaction that accompany the successful completion of any task would be incomplete without the mention of those who made it possible. We are glad to express our gratitude towards our prestigious institution **AMC ENGINEERING COLLEGE** for providing us with utmost knowledge, encouragement and the maximum facilities in undertaking this project.

We wish to express sincere thanks to our respected chairman **Dr. K. R. Paramahamsa** and beloved principal **Dr. Rajesh S** for all their support.

We express our deepest gratitude and special thanks to **Dr. K Ananthapadmanabha**, **H.O.D, Dept. Of Computer Science Engineering**, for all her guidance and encouragement.

We sincerely acknowledge the guidance and constant encouragement of our mini- project guide, **Mrs. Veena Bhat , Assistant Prof. , Dept. of Computer Science Engineering.**

**RAHUL MISHRA [1AM19CS159]**

**RAHUL GOOGIKOLL [1AM19CS158]**

**SAURAV KUMAR [1AM19CS195]**

**OM PRAKASH JHA [1AM19CS133]**

**D KARTHIK REDDY [1AM20CS401]**

# ABSTRACT

The entitled project “**CAR DEALERSHIP DATABASE MANAGEMENT SYSTEM**” is made keeping in mind all the aspects of the Car dealerships. By all the aspects I mean, it will be capable of doing all the necessary operations/functions that are done in any Car dealership for example- reservation of a car, booking of a car ,keeping track of customer details

,billing details etc. Since all the work that is to be done by this software can also be done manually, but this consumes time and labour. So this software will be a relief to those who have to do all this work manually. The knowledge of computers and programming has become a basic skill needed to survive in present information based on society.

The motive to make this project is to make such kind of software which is very easy to use. There will not be need of any training and the person who does not have much knowledge of computers can also use this .Through this project the details of the customers that have booked cars can be retrieved if necessary. All the records of the customers will be kept for further enquiries.

# CONTENTS

## Title Page No

**INTRODUCTION 7**

* Project Outline
* Project Goals

**SYSTEM SPECIFICATION 8**

* Hardware requirements
* Software requirements

**DESIGN 9**

**IMPLEMENTATION AND CODING 13**

* User defined functions
* Source code

**SNAPSHOTS. 33**

**CONCLUSION 37**

**REFERENCES 37**

**Chapter 1**

**INTRODUCTION**

**1.1 Project Outline**

This application is going to be used by:

1. Employees

2. Customer

This application is meant for users (Employees and Customers) who can access the information from database about car deals. Our objective is so that executives and administrators can easily automate their work with less effort. For example they can generate quotations for any enquiry for a car in no time.

They can also have the customer’s feedback to the show room authority regarding show room services.

**1.2 Project Goals**

The System should be capable of performing the following:

* Store basic information regarding cars, employees, customers, accessories and services provided by the organization.
* Store salary information of employees (entered by the team leaders in each city) such as, working hours, salary per hour, salary before tax, tax percentage, total amount of tax paid , salary after tax, social security fee, on monthly basis
* Help employees automate the billing process by quickly fetching car/accessory details from the database.

* System should be able to generate invoices and maintain transaction details.
* Should be able to list the inventory of cars currently available in a dealership.
* Should help customers to access car details like model, make, engine no., price etc.
* Help the organization to maintain customer details like customer name, transaction/purchase details, car purchased, purchase medium, purchase date etc.
* Display details of various dealerships of the same franchise/organization like, profits, dealership name, number of cars in dealership, number of employees etc.

**Chapter 2**

**SYSTEM SPECIFICATIONS**

**2.1 Hardware Requirements**

Minimum Requirements:

* Processor: Intel Pentium 4/ AMD Athlon series
* RAM: 512MB
* Storage: 100MB

Recommended Requirements:

* Processor: Intel 11th generation I-series/AMD Ryzen Threadripper series
* RAM: 8GB
* Storage: 500GB

**2.2 Software Requirements**

OS:Windows,GNU/Linux Distributions, Mac OS, BSD, 64-bit

**2.3 Tech Stack**

**Frontend**

* **HTML 5:**

HTML5 is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and last major HTML version that is a World Wide Web Consortium recommendation. The current specification is known as the HTML Living Standard.

* **CSS 3:**

CSS3 is the latest version of the CSS specification. CSS3 adds several new styling features and improvements to enhance the web presentation capabilities. Note: Our CSS tutorial will help you to learn the fundamentals of the latest CSS3 language, from the basic to advanced topics step-by-step.

* **JavaScript:**

JavaScript, often abbreviated JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. Over 97% of websites use JavaScript on the client side for web page behavior, often incorporating third-party libraries.

**Backend**

* **Django:**

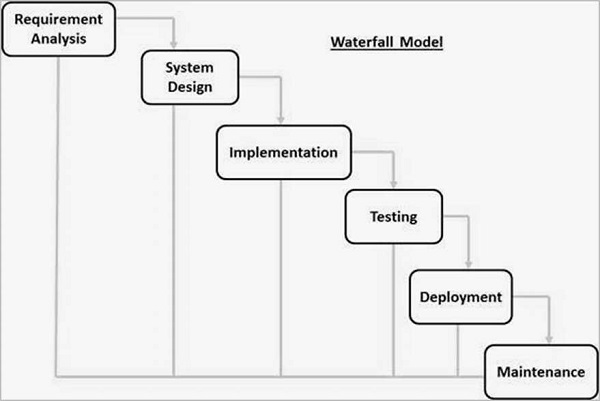
Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source.

* **SQLite3:**
* SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine. SQLite is the most used database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day.

**Chapter 3**

**DESIGN**

**3.1 Development model**



* **Requirement Analysis:**

All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.

* **Design stage:**

The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

* **Implementation:**

With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

* **Integration and Testing:**

All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

* **Deployment of System:**

Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.

* **Maintenance:**

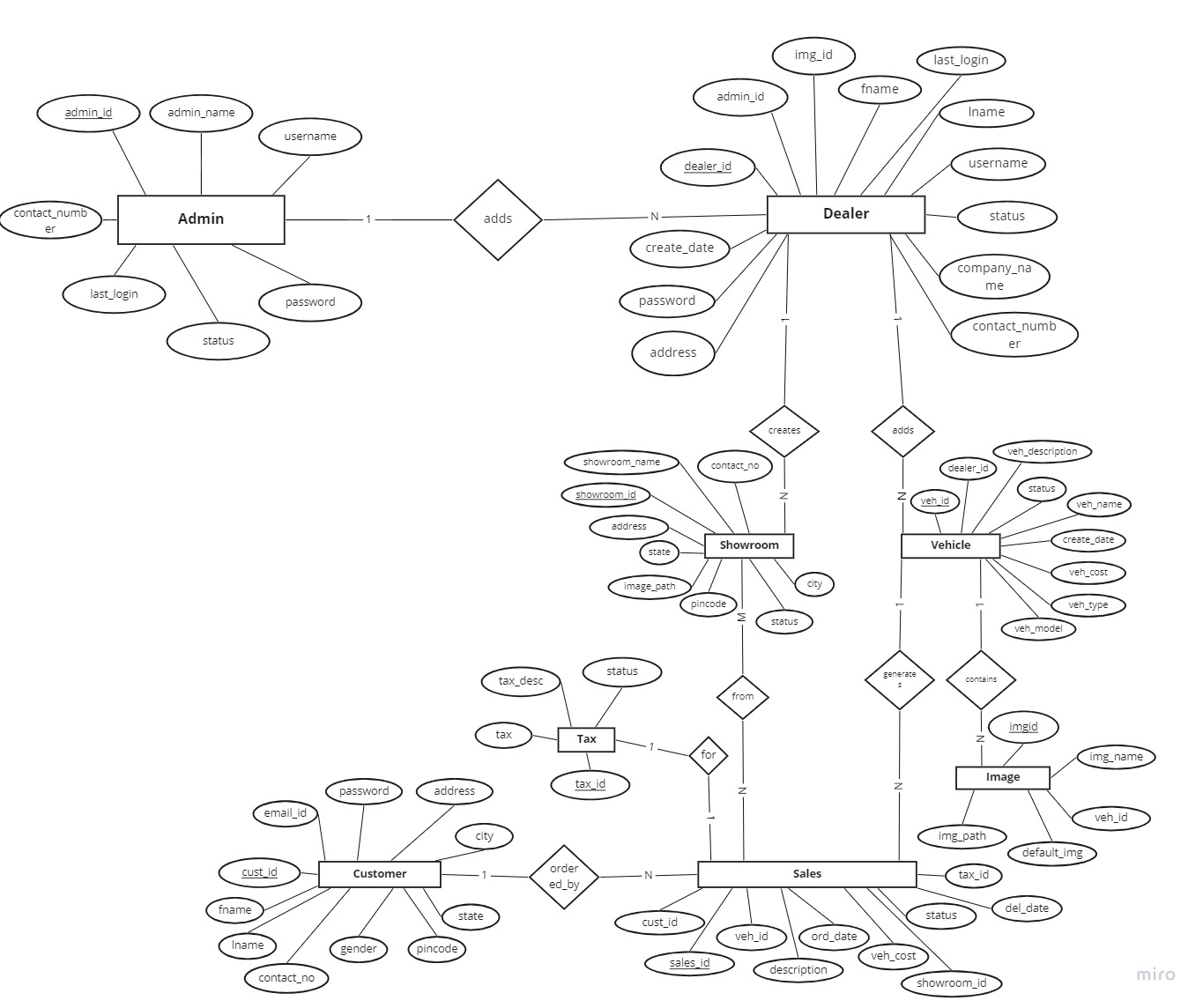
There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards ,

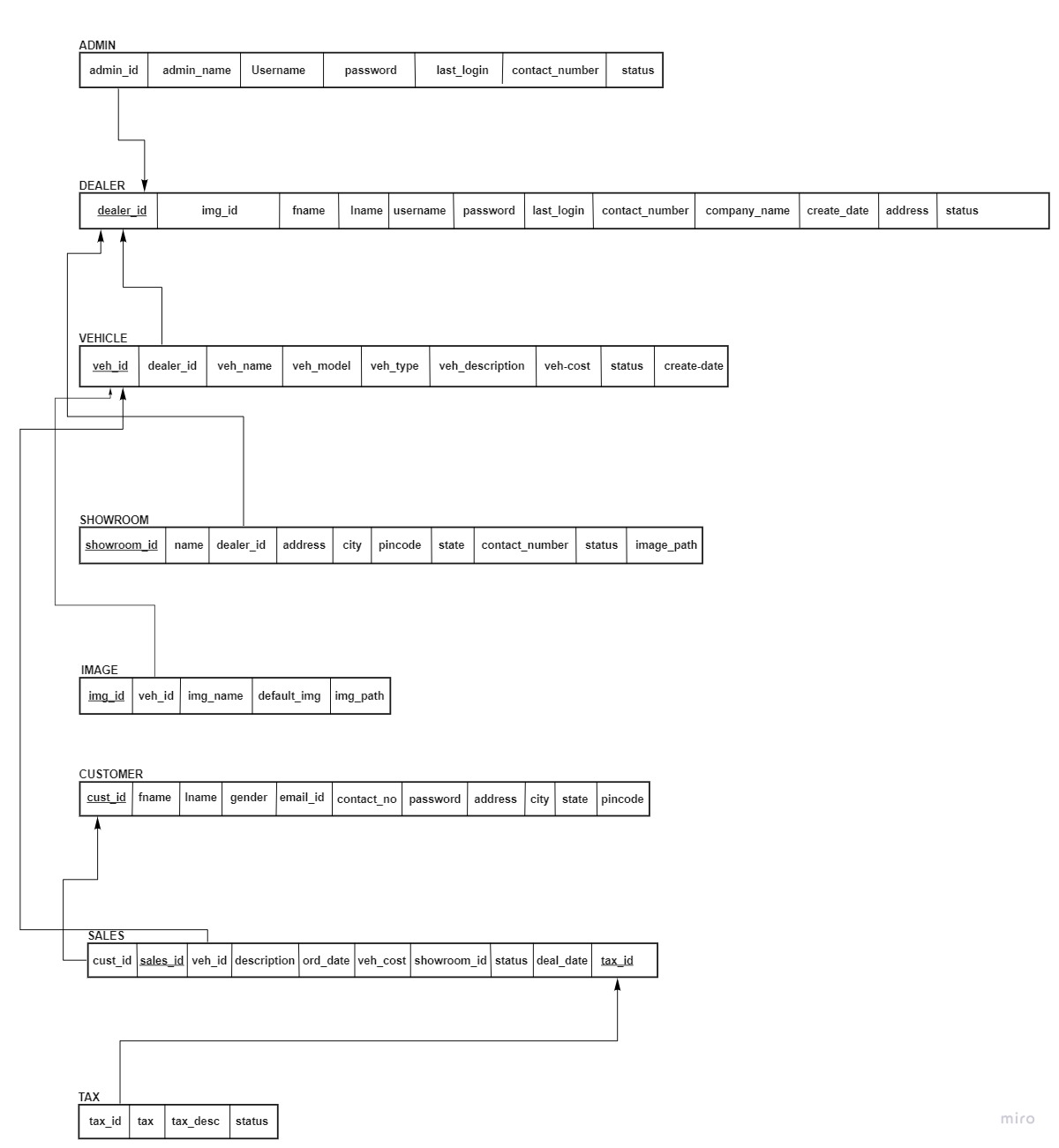
(like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model, phases do not overlap.

**3.2 Database Description**

**ER-Diagram**



**Schema Diagram**

****

**Chapter 4**

**IMPLEMENTATION AND CODING**

**4.1 Backend Models(Tables)**

## **Dealer Model**

Contains the Dealer table definition and attribute list, based and derived from the schema diagram

**# src/models.py**

**class Dealer(models.Model) => (attrs)**

The Dealer Table hereby referred to as the "Dealer Object" further into this section of the documentation consists of the following attributes

**1.) dealer\_id :**

Attribute definition:

**(variable) dealer\_id: IntegerField => (**

**primary\_key: bool,**

**unique: bool**

**)**

Source code:

**dealer\_id = models.IntegerField(**

**primary\_key=True,**

**unique=True**

**)**

Refers to the identification number of the dealer , **primary\_key=True** parameter is setting the attribute to a primary key. The **unique=True** parameter is useful in referencing this particular attribute via ***Foreign Keys***.

**2.) dealer\_img:**

Attribute definition:

**(variable) dealer\_img: ImageField => (**

**blank: bool**

**)**

Source code:

**dealer\_img = models.ImageField(**

**blank=False**

**)**

Refers to the dealer's image field the **ImageField** method stores uploaded image in the room **/media/** folder. **blank = False**

## **3.) dealer\_name:**

## Attribute definition:

**(variable) dealer\_name: CharField =>(**

**max\_length: int,**

**blank: bool, null: bool**

**)**

Source code:

**dealer\_name = models.CharField(**

**max\_length=100,**

**blank=False,**

**null=False**

**)**

Refers to the dealer's name field the **NameField** method stores the name of the dealer class, **null=False** parameter dealer class,null=Falseparameter setting to the null attribute.max\_length is set to (100 and it's restricts the number of characters used by the user by (100. **Blank=False**

## **4.) dealer\_username**

## Attribute definition:

**(variable) dealer\_username: ForeignKey => (**

**(class) User,**

**(parameter) on\_delete: (...) -> None,**

**(parameter) related\_name: str | None**

**)**

Source code:

**dealer\_username = models.ForeignKey(**

**User, on\_delete=models.CASCADE,**

**related\_name='d\_username'**

**)**

refers to the username of the dealer, as apparent here this attribute is a foreign key that refers to **(class) User** , it has a parameter **(parameter) on\_delete: (...) -> None** set to **models.CASCADE** which means on deletion of a record from

this table it reflects by comitting deletion on the referenced table as well. The **(parameter) related\_name: str | None** parameter sets up an alternate name or view to avoid arising naming conflicts.

**5.) last\_login:**

Attribute definition:

**(variable) last\_login: DateTimeField => (**

**(parameter) auto\_now: bool**

**)**

Source code:

**last\_login = models.DateTimeField(auto\_now=True)**

The **last\_login** attribute refers to the last time the dealer had logged in. the **(parameter) auto\_now: bool** is set to **True**, the **auto\_now** parameter automatically sets the login time based on server timezone.

**6.) contact:**

Attribute definition:

**(variable) contact: IntegerField => (**

**null: bool,**

**blank: bool**

**)**

Source code:

**contact = models.IntegerField(**

**null=False,**

**blank=False**

**)**

refers to the contact details of the dealer class.the **(parameter)null: bool** is set to **False, Blank=False**

**7.) company\_name:**

Attribute definition:

**(variable) company\_name: CharField => (**

**(parameter) max\_length: int | None,**

**(parameter) null: bool,**

**(parameter) blank: bool**

**)**

Source code:

**company\_name = models.CharField(**

**max\_length=100,**

**null=False,**

**blank=False**

**)**

The **company name** attribute refers to the name of the company that the dealer belongs to, since it is a **CharField** it consists of the **(parameter) max length** that is set to **100** . This field cannot be left **null** or **blank**.

**8.) address:**

Attribute definition:

**(variable) address: TextField => (**

**(parameter) max\_length: int | None,**

**(parameter) null: bool,**

**(parameter) blank: bool**

**)**

Source code:

**address = models.TextField(**

**max\_length=300,**

**blank=False,**

**null=False**

**)**

The **Address** attribute refers to the Dealer's address, **(parameter) max\_length:**

**int |** **None** is set to **300**, and **(parameter) blank: bool** and **(parameter) null: bool** are set to **False**.

**9.) Created:**

Attribute definition:

**(variable) created: DateTimeField => (**

**(parameter) auto\_created: bool**

**)**

Source code:

**created = models.DateTimeField(**

**auto\_created=True**

**)**

The **created** attribute refers to the date the dealer's profile was created,

**(parameter) auto\_created: bool** is set to **True**, this is a method passed in as a parameter which sets the value to current date and time based on the server region and timezone,

**10.) status:**

Attribute definition:

**(variable) status: CharField =>(**

**(parameter) max\_length: int | None,**

**(parameter) choices: \_FieldChoices | None,**

**(parameter) default: Any**

**)**

**here the choices are defined in a list of choice sets,**

**(constant) STATUS\_CHOICE: list**

Source code:

**STATUS\_CHOICE = [**

**('Active', 'Active'),**

**('Inactive', 'Inactive')**

**]**

**status = models.CharField(**

**max\_length=20,**

**choices=STATUS\_CHOICE,**

**default= 'Active'**

**)**

refers to the status in the dealer class.Since it is a **CharField** it consists of the **(parameter) max\_length** that is set to **20**. By **default= Any** the

**STATUS CHOICE** is set to **Active.**

**Vehicle Model**

Contains the Vehicle table definition and attribute list, based and derived from the schema diagram

**1.) vehicle\_id:**

Attribute definition:

**(variable) vehicle\_id: IntegerField => (**

**(parameter) primary\_key: bool,**

**(parameter) unique: bool**

**)**

Source code:

**vehicle\_id = models.IntegerField(**

**primary\_key=True,**

**unique=True**

**)**

The **vehicle\_id** attribute refers to the numeric id of the vehicle entries present in the table, it is a **primary key** and it is set to be **unique**.

**2.) dealer\_id:**

Attribute definition:

**(variable) dealer\_id: ForeignKey =>(**

**(class) Dealer,**

**(parameter) on\_delete: (...) -> None,**

**(parameter) to\_field: str | None**

**)**

Source code:

**dealer\_id = models.ForeignKey(**

**Dealer,**

**to\_field='dealer\_id',**

**on\_delete=models.CASCADE**

**)**

refers to the id of the dealer, as apparent here this attribute is a foreign key that refers to **(class) dealer**, it has a parameter **(parameter) on delete: (…) -> None** set to **models.CASCADE** which means on deletion of a record from this table it reflects by comitting deletion on the referenced table as well. The **to field** is set to

**dealer id**

**3.) vehicle\_img:**

Attribute definition:

**(variable) vehicle\_img: ImageField => (**

**blank: Any**

**)**

Source code:

**vehicle\_img = models.ImageField(**

**blank=False**

**)**

The **vechicle\_img** attribute contains the vehicle images uploaded to the server stored at **/media/** folder in the **MEDIA DIR** directory of the server, **blank** is set to **False** so an image is required for every record uploaded.

**4.) name:**

Attribute definition:

**(variable) name: CharField => (**

**(parameter) max\_length: int | None,**

**(parameter) blank: bool**

**)**

Source code:

**name = models.CharField(**

**max\_length=100,**

**blank=False**

**)**

The **name** attribute refers to the vehicle name, it's **max\_length** is set to **100** and **blank** is **False** so it cannot be left empty.

**5.) type:**

Attribute definition:

**(variable) type: CharField =>(**

**(parameter) max\_length: int | None,**

**(parameter) choices: \_FieldChoices | None,**

**(parameter) blank: bool**

**)**

here the choices are defined in a list of choice sets,

**(constant) VEHICLE\_TYPE: list**

**VEHICLE\_TYPE = [**

**("Hatchback", "Hatchback"),**

**("Sedan", "Sedan"),**

**("MPV", "MPV"),**

**("SUV", "SUV"),**

**("Crossover", "Crossover"),**

**("Coupe", "Coupe"), ("Convertible", "Convertible")**

**]**

Source code:

**type = models.CharField**

**( max\_length=100,**

**choices=VEHICLE\_TYPE,**

**blank=False**

**)**

The **type** attribute refers to the the vehicle type,it's **max\_length** is set **100** and **blank=False** so it cannot be left empty. The **choice** parameter is set to **VECHICLE TYPE**

**6.) Description:**

Attribute definition:

**(variable) description: TextField => (**

**(parameter) max\_length: int | None,**

**(parameter) blank: bool**

**)**

Source code:

**description = models.TextField(**

**max\_length=100,**

**blank=False**

**)**

The **description** attribute refers to vehicle description, it's **max\_length** is **200** and **blank** is set to **False**.

**7.) cost:**

Attribute definition:

**(variable) cost: IntegerField => (**

**(parameter) blank: bool**

**)**

Source code:

**cost = models.IntegerField(**

**blank=False**

**)**

The **cost** attribute refers to the vehicle cost, it cannot be **blank** since blank is set to **False**.

**8.) status:**

Attribute definition:

**(variable) status: CharField => (**

**(parameter) max\_length: int | None,**

**(parameter) choices: \_FieldChoices | None,**

**(parameter) blank: bool**

**)**

here the choices are defined in a list of choice sets,

**(constant) VEH\_STATUS: list**

**VEH\_STATUS = [**

**("For Sale", "For Sale"),**

**("Sold", "Sold")**

**]**

Source code:

**status = models.CharField(**

**max\_length=20,**

**choices=VEH\_STATUS,**

**blank=False,**

**default='For Sale'**

**)**

The **status** attribute refers to the vehicle sale status, the choice sets are defined in **VEH STATUS**, and **default** choice in **For sale**.

**9.) created:**

Attribute definition:

**(variable) created: DateTimeField => (**

**(parameter) auto\_now\_add: bool**

**)**

Source code:

**created = models.DateTimeField(**

**auto\_now\_add=True**

**)**

The **created** attributes refers to the date on which the vehicle was created. The

**auto now add:bool** is set to **True**.

**Customer Model**

Contains the Customer table definition and attribute list, based and derived from the schema diagram

**1.) customer\_id:**

Attribute definition:

**(variable) customer\_id: IntegerField => (**

**(parameter) primary\_key: bool,**

**(parameter) unique: bool**

**)**

Source code:

**customer\_id = models.IntegerField(**

**primary\_key=True,**

**unique=True**

**)**

The **customer\_id** attribute refers to numeric customer id, it is a **primary key**, and it is **unique**.

**2.) first\_name:**

Attribute definition:

**(variable) first\_name: CharField => (**

**(parameter) max\_length: int | None,**

**(parameter) blank: bool**

**)**

Source code:

**first\_name = models.CharField(**

**max\_length=100,**

**blank=False**

**)**

The **first\_name** attribute stores the customer first name.The **max\_length**, is set to **100**. It cannot be left **blank**

**3.) last\_name:**

Attribute definition:

**(variable) last\_name: CharField =>(**

**(parameter) max\_length: int | None,**

**(parameter) blank: bool**

**)**

Source code:

**last\_name = models.CharField(**

**max\_length=100,**

**blank=False**

**)**

The **last\_name** attribute refers customer's last name, it cannot be left **blank**.

**4.) Gender:**

Attribute definition:

**(variable) gender: CharField =>(**

**(parameter) max\_length: int | None,**

**(parameter) blank: bool,**

**(parameter) choices: \_FieldChoices | None**

**)**

here the choices are defined in a list of choice sets,

**(constant) GENDER: list**

**GENDER = [**

**("Male", "Male"),**

**("Female", "Female"),**

**("Other", "Other")**

**]**

Source code:

**gender = models.CharField(**

**max\_length=100,**

**blank=False,**

**choices=GENDER**

**)**

The **gender** attribute refers to the customer's gender, the gender field has **choice** defined in a list of sets defined in **GENDER**.

**5.) email:**

Attribute definition:

**(variable) email: CharField => (**

**(parameter) max\_length: int | None,**

**(parameter) blank: bool**

**)**

Source code:

**email = models.CharField(**

**max\_length=100,**

**blank=False**

**)**

The **email** attribute refers to the customer's email.it's **max\_length** is set to **100** and it cannot be left **blank**.

**6.) Contact:**

Attribute definition:

**(variable) contact: IntegerField =>(**

**(parameter) blank: bool**

**)**

Source code:

**contact = models.IntegerField(**

**blank=False**

**)**

The **contact** attribute refers to the contact information of the customer, it cannot be left **blank**.

**7.) city:**

Attribute definition:

**(variable) city: CharField => (**

**(parameter) max\_length: int | None,**

**(parameter) blank: bool**

**)**

Source code:

**city = models.CharField(**

**max\_length=100,**

**blank=False**

**)**

The **city** attribute refers to the customer's city name in which he/she resides. It’s **max\_length** is set to **100** and it cannot be left **blank**.

**8.) State:**

Attribute definition:

**(variable) state: CharField =>(**

**(variable) state: CharField,**

**(parameter) blank: bool**

**)**

Source code:

**state = models.CharField(**

**max\_length=100,**

**blank=False**

**)**

The **state** attribute refers to the customer's state, it cannot be **blank**.

**9.) Address:**

Attribute definition:

**(variable) address: TextField => (**

**(parameter) max\_length: int | None**

**)**

Source code:

**address = models.TextField(**

**max\_length=100,**

**blank=False**

**)**

The **address** attribute refers to the customer’s address, it cannot be left **blank**.

**Sale Model**

Contains the Sale table definition and attribute list, based and derived from the schema diagram

**1.) customer\_id:**

Attribute definition:

**variable) customer\_id: ForeignKey => (**

**(class) Customer,**

**(parameter) to\_field: str | None,**

**(parameter) on\_delete: (...) -> None**

**)**

Source code:

**customer\_id = models.ForeignKey(**

**Customer,**

**to\_field='customer\_id',**

**on\_delete=models.CASCADE**

**)**

The **customer\_id** attribute refers to the numeric id of the referenced customer.

**2.) sale\_id:**

Attribute definition:

**(variable) sale\_id: IntegerField =>(**

**(parameter) primary\_key: bool,**

**(parameter) unique: bool**

**)**

Source code:

**sale\_id = models.IntegerField(**

**primary\_key=True,**

**unique=True**

**)**

The **sale\_id** attribute refers to the numeric id of a sale made.

**3.) vehicle\_id:**

Attribute definition:

**(variable) vehicle\_id: ForeignKey =>(**

**(class) Vehicle,**

**(parameter) to\_field: str | None,**

**(parameter) on\_delete: (...) -> None**

**)**

Source code:

**vehicle\_id = models.ForeignKey(**

**Vehicle,**

**to\_field='vehicle\_id',**

**on\_delete=models.CASCADE**

**)**

The **vehicle\_id** attribute refers to the numeric id of the referenced vehicle object.

**4.) Description:**

Attribute definition:

**(variable) description: TextField =>(**

**(parameter) max\_length: int | None,**

**(parameter) blank: bool**

**)**

Source code:

**description = models.TextField(**

**max\_length=100,**

**blank=False**

**)**

The **description** attribute describes the sale made.

**5.) order\_date:**

Attribute definition:

**(variable) order\_date: DateTimeField =>(**

**(variable) order\_date: DateTimeField**

**)**

Source code:

**order\_date = models.DateTimeField(**

**auto\_created=True**

**)**

The **order\_date** attribute refers to the date the sale was made.

**6.) Cost:**

Attribute definition:

**(variable) cost: IntegerField => (**

**(parameter) blank: bool**

**)**

Source code:

**cost = models.IntegerField(**

**blank=False**

**)**

The **cost** refers to the cost paid by the customer to the dealership.

**7.) deal\_date:**

Attribute definition:

**(variable) deal\_date: DateTimeField => (**

**(parameter) auto\_created: bool**

**)**

Source code:

**deal\_date = models.DateTimeField(**

**auto\_created=True**

**)**

The **deal\_date** attribute refers to the date the car deal was made.

**8.) Status:**

Attribute definition:

**(variable) status: CharField => (**

**(parameter) max\_length: int | None,**

**(parameter) blank: bool,**

**(parameter) choices: \_FieldChoices | None**

**)**

here the choice list is given in a list of sets,

**(constant) TAX\_STATUS: list**

**SALE\_STATUS = [**

**("Sold", "Sold"),**

**("On hold", "On hold"),**

**("Rejected", "Rejected")**

**]**

Source code:

**status = models.CharField(**

**max\_length=20,**

**blank=False,**

**choices=SALE\_STATUS**

**)**

The **status** attribute refers to the deal status.

**9.) tax\_id:**

Attribute definition:

**(variable) tax\_id: IntegerField =>(**

**(parameter) unique: bool,**

**(parameter) blank: bool**

**)**

Source code:

**tax\_id = models.IntegerField(**

**unique=True,**

**blank=False**

**)**

The **tax** attributes refers to the id of the tax paid for the vehicle.the **unique** is set to **True** and it cannot be left blank.

**Tax Model**

Contains the Tax table definition and attribute list, based and derived from the schema diagram

**1.) tax\_id:**

Attribute definition:

**(variable) tax\_id: ForeignKey => (**

**(class) Sale,**

**(parameter) to\_field: str | None,**

**(parameter) on\_delete: (...) -> None**

**)**

Source code:

**tax\_id = models.ForeignKey(**

**Sale,**

**to\_field='tax\_id',**

**on\_delete=models.CASCADE**

**)**

**The tax\_id** attributerefers to the numeric id referenced in the Sale

Model/Table.

**2.) tax:**

Attribute definition:

**(variable) tax: IntegerField =>(**

**(variable) tax: IntegerField**

**)**

Source code:

**tax = models.IntegerField(**

**blank=False**

**)**

The **tax** attribute refers to the amount of tax applied to the bill of sale.

**3.) Description:**

Attribute definition:

**(variable) description: TextField=>(**

**(parameter) max\_length: int | None,**

**(parameter) blank: bool**

**)**

Source code:

**description = models.TextField(**

**max\_length=100, blank=False**

**)**

The **description** attribute refersto the description of the tax paid.

**4.) status:**

Attribute definition:

**(variable) status: CharField => (**

**(parameter) max\_length: int | None,**

**(parameter) blank: bool,**

**(parameter) choices: \_FieldChoices | None**

**)**

here the choice list is given as,

**(constant) TAX\_STATUS: list**

**TAX\_STATUS = [**

**("Approved", "Approved"),**

**("Pending", "Pending"),**

**("Rejected", "Rejected")**

Source code:

**status = models.CharField(**

**max\_length=20,**

**blank=False,**

**choices=TAX\_STATUS**

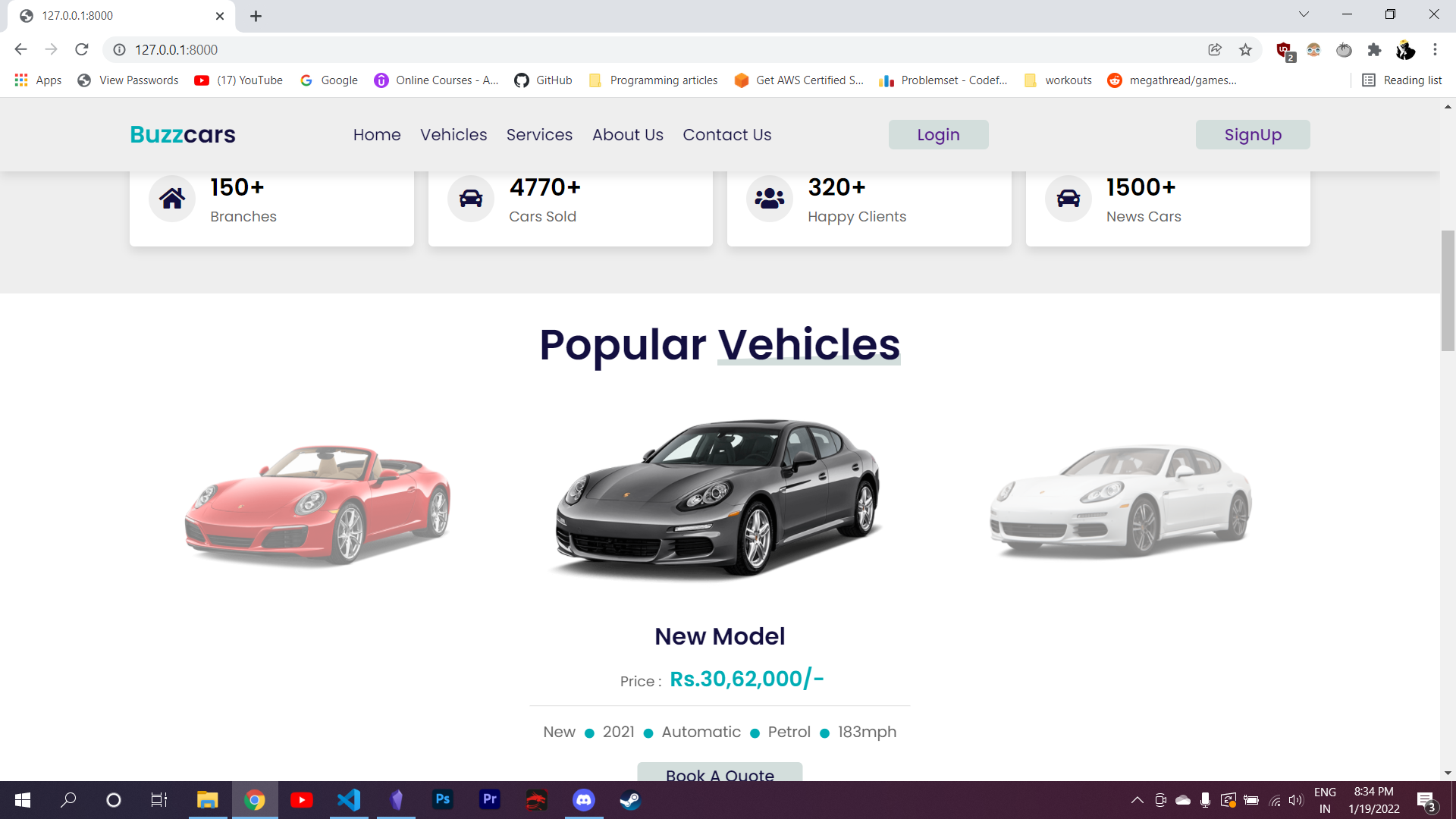
**)**

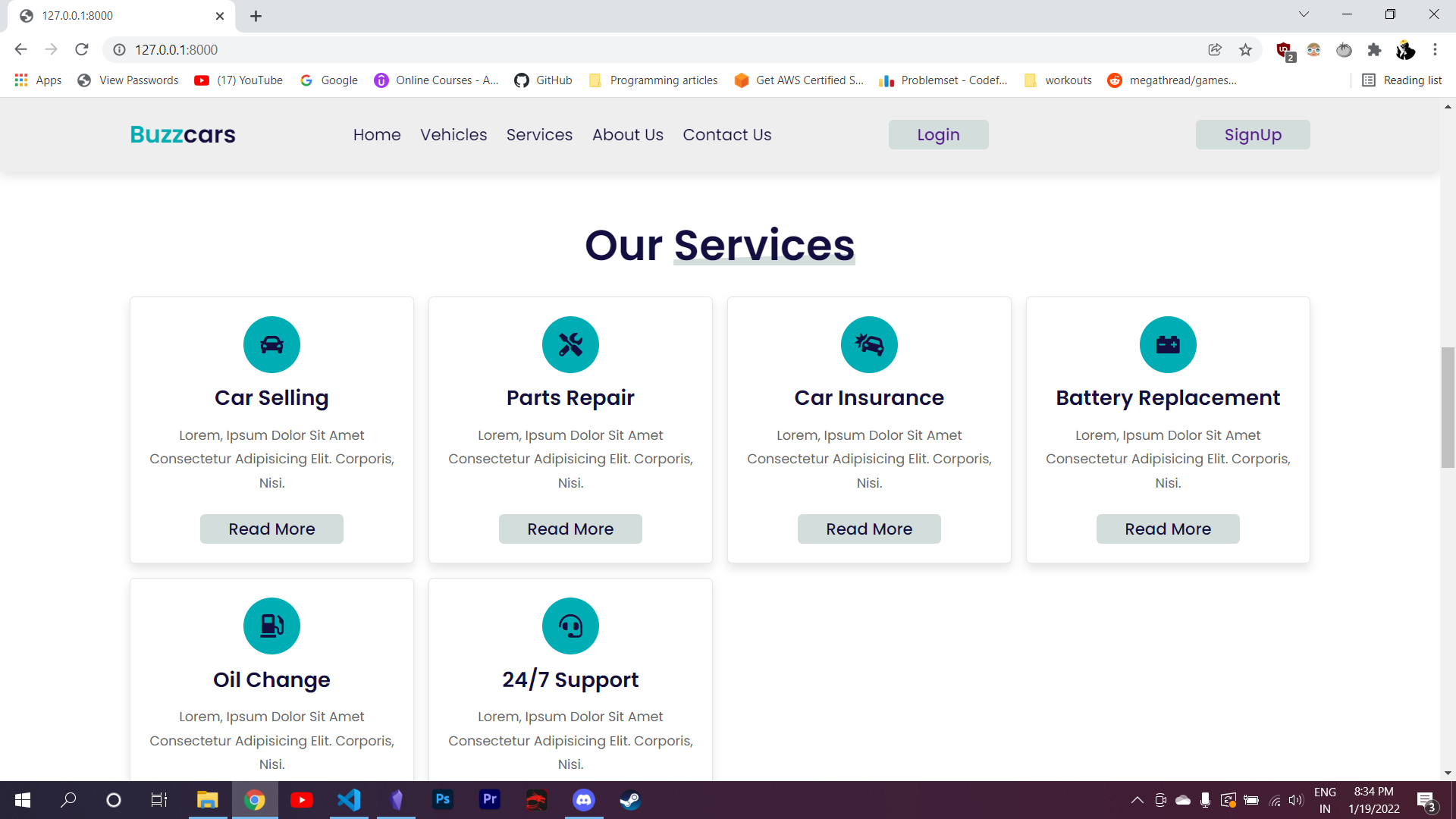
The status attribute refers to the status of the tax payment, it refers to TAX STATUS list for choice sets.

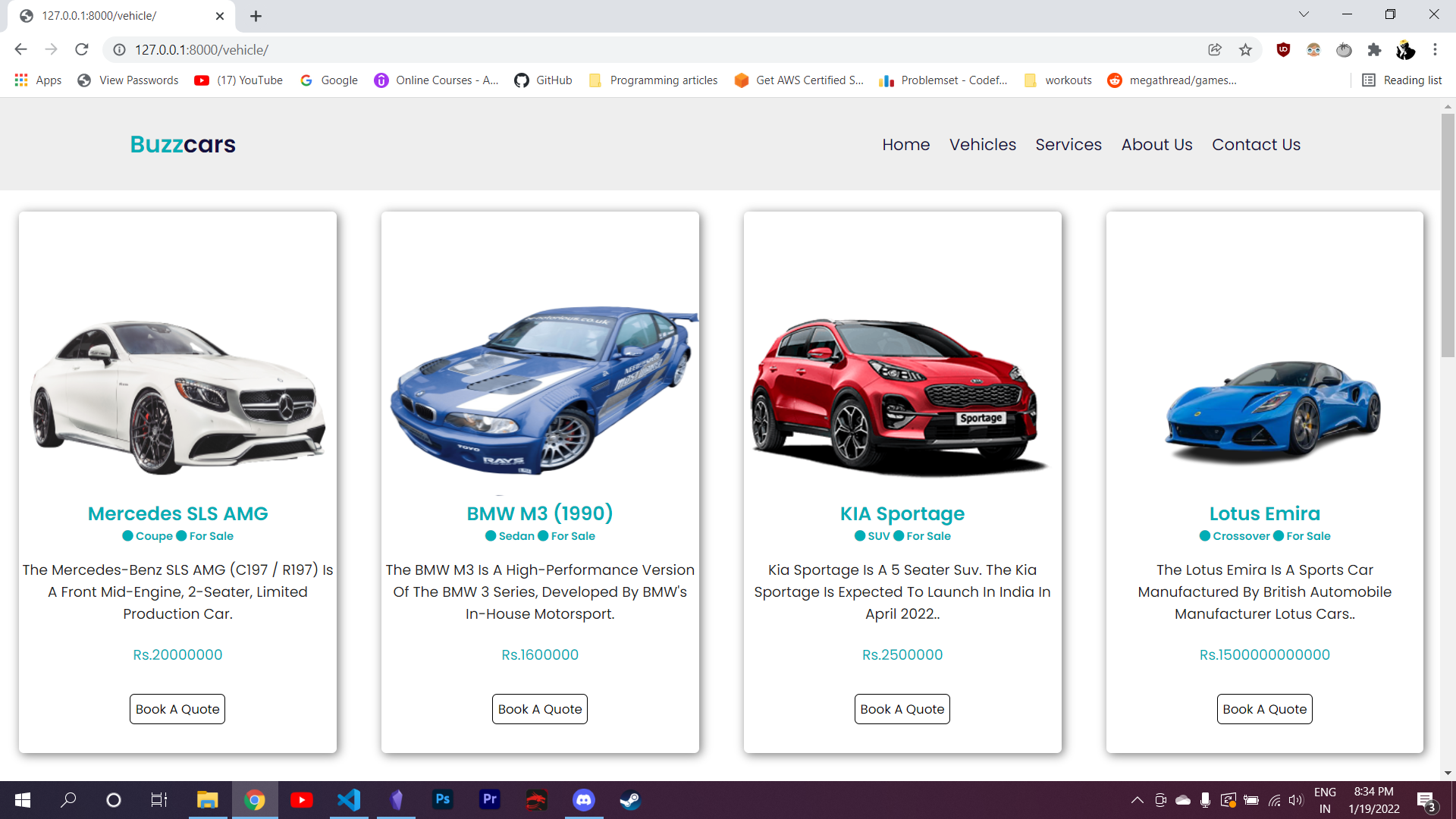
**Chapter 5**

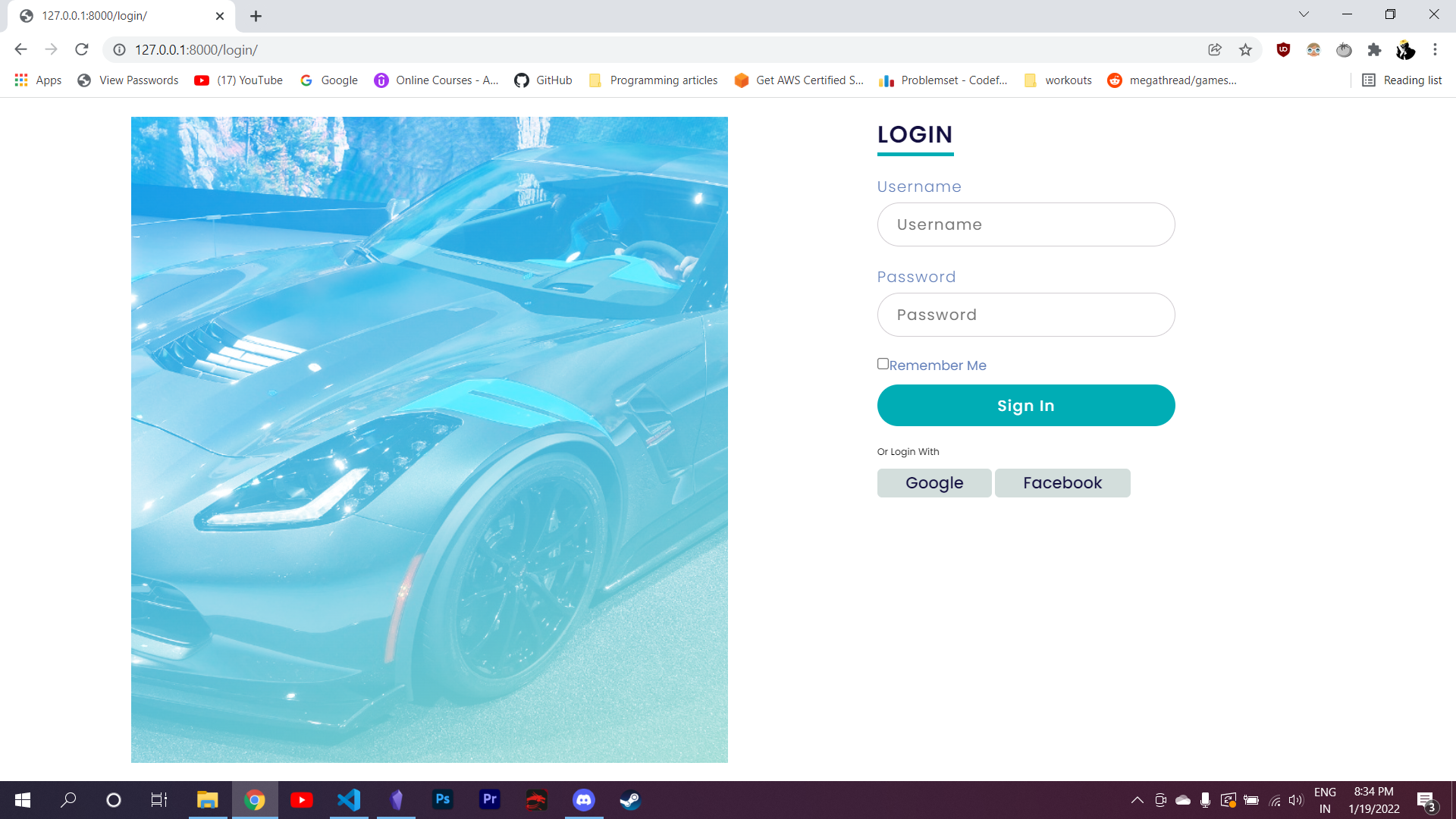
**SNAPSHOTS**

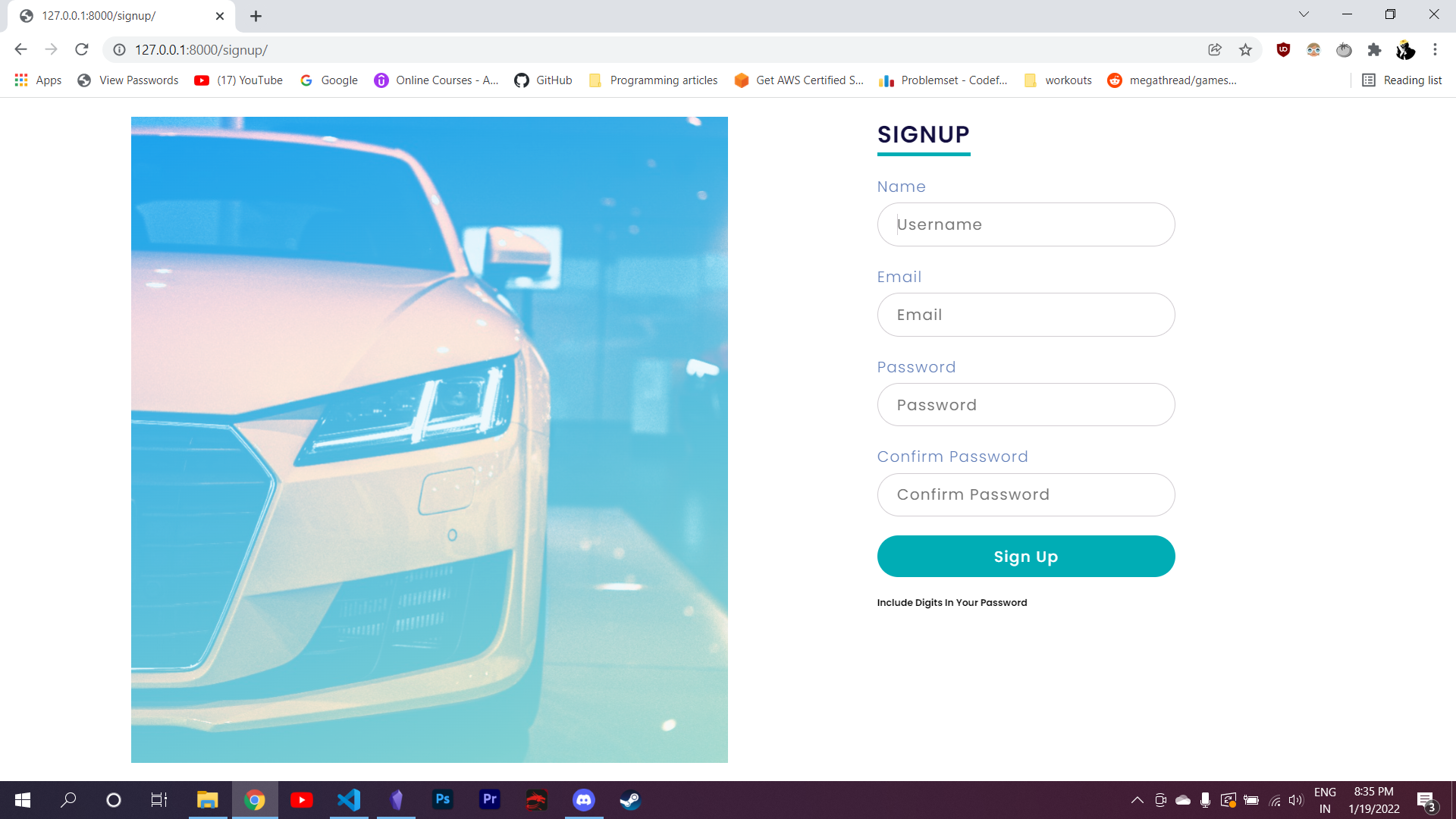


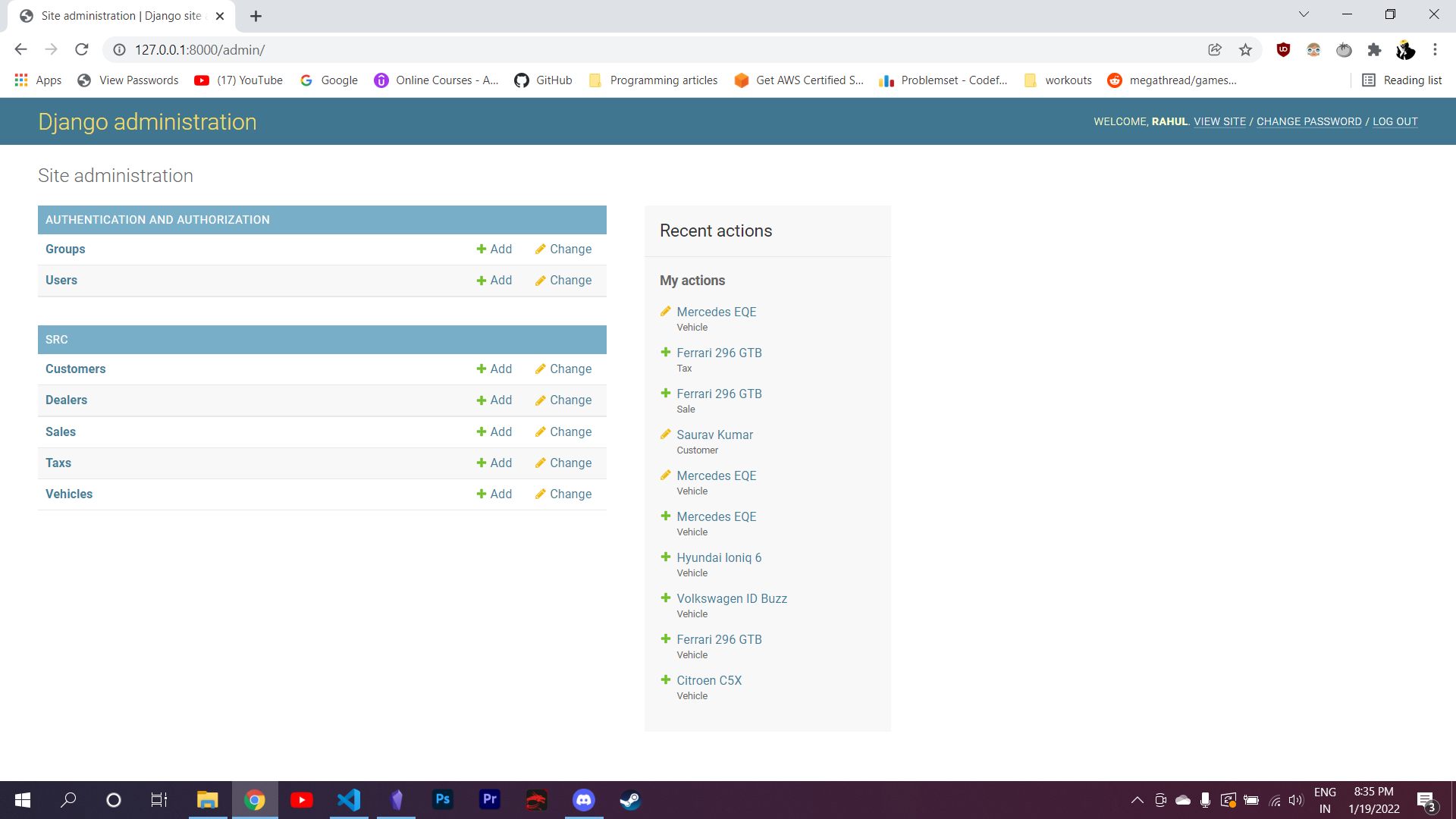












**Chapter 6**

**CONCLUSION**

The project is the perfect system for the car showrooms to manage their data and help their customers to buy a car without any problem. It makes the process of buying a car very simpler.

In future, all the updates about the cars will be centralized because of the application being online.

Provision for customer to access our system will be available.

Users will be able to visit our show room being at home due to the system being online.

**REFERENCES**

* www.Google.com
* https://www.djangoproject.com/
* www.w3schools.com
* https://www.sqlite.org/docs.html