

# **VISVESVARAYA TECHNOLOGICAL UNIVERSITY BELGAUM-590014**



## **A Mini-Project Report On “CAR DEALERSHIP DATABASE SYSTEM”**

*A Mini-project report submitted in partial fulfilment 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)**

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-560083  
2021-2022**

# **Team Members**

1. Rahul Googikoll [1AM19CS158]
2. Rahul Mishra [1AM19CS159]
3. Saurav Kumar [1AM19CS195]
4. Om Prakash Jha [1AM19CS133]

## **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.

# 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

### Tech stack used:

**Front end:** HTML5, CSS, JS

- **HTML5:**

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:**

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.

- **JS:**

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.

## **Back end: Django, SQLite3**

- **Django:**

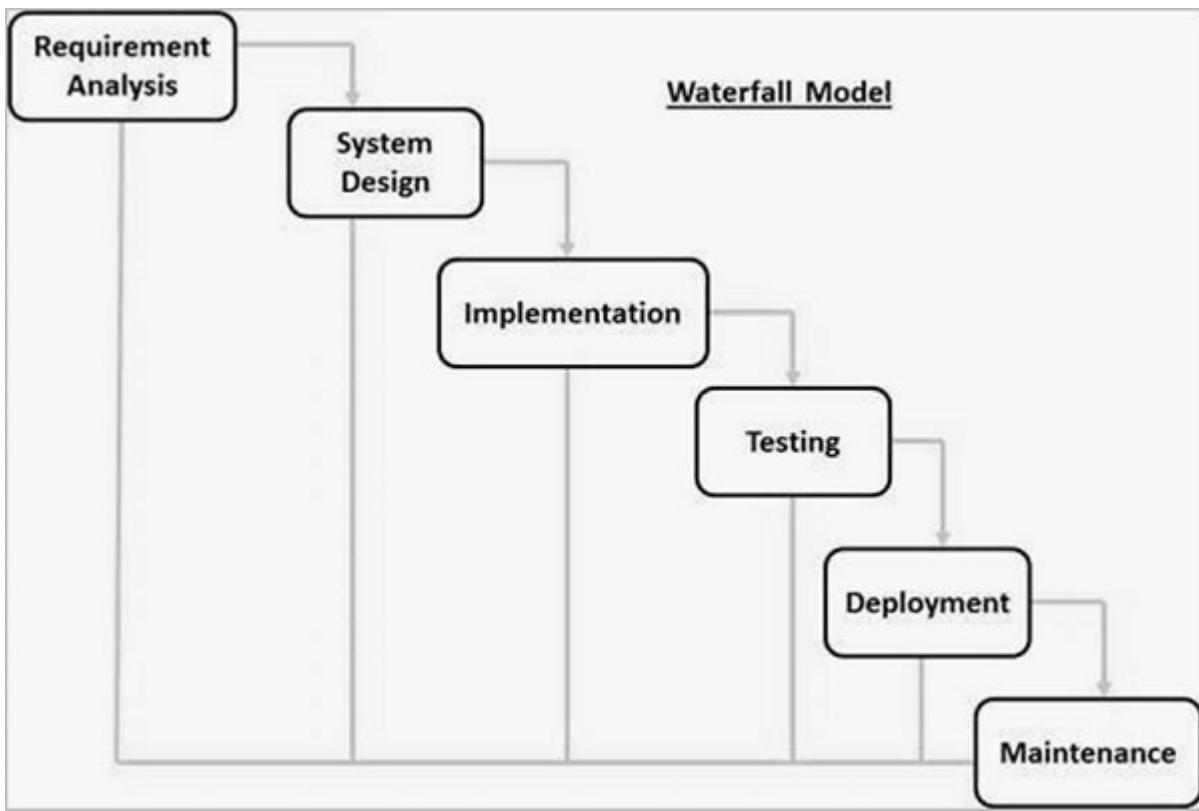
Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

- **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.

# 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.

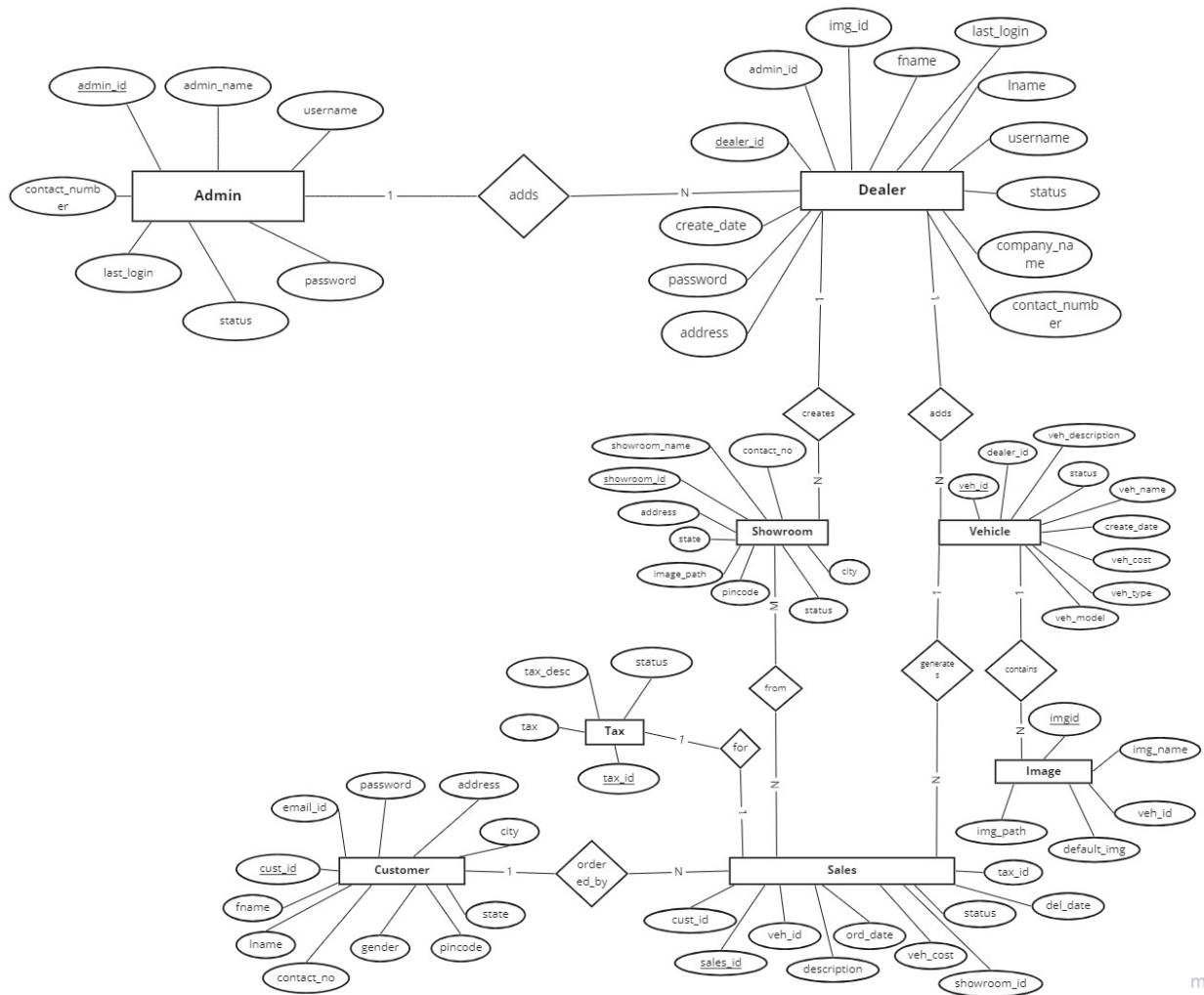
- **Maintenance:**

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.

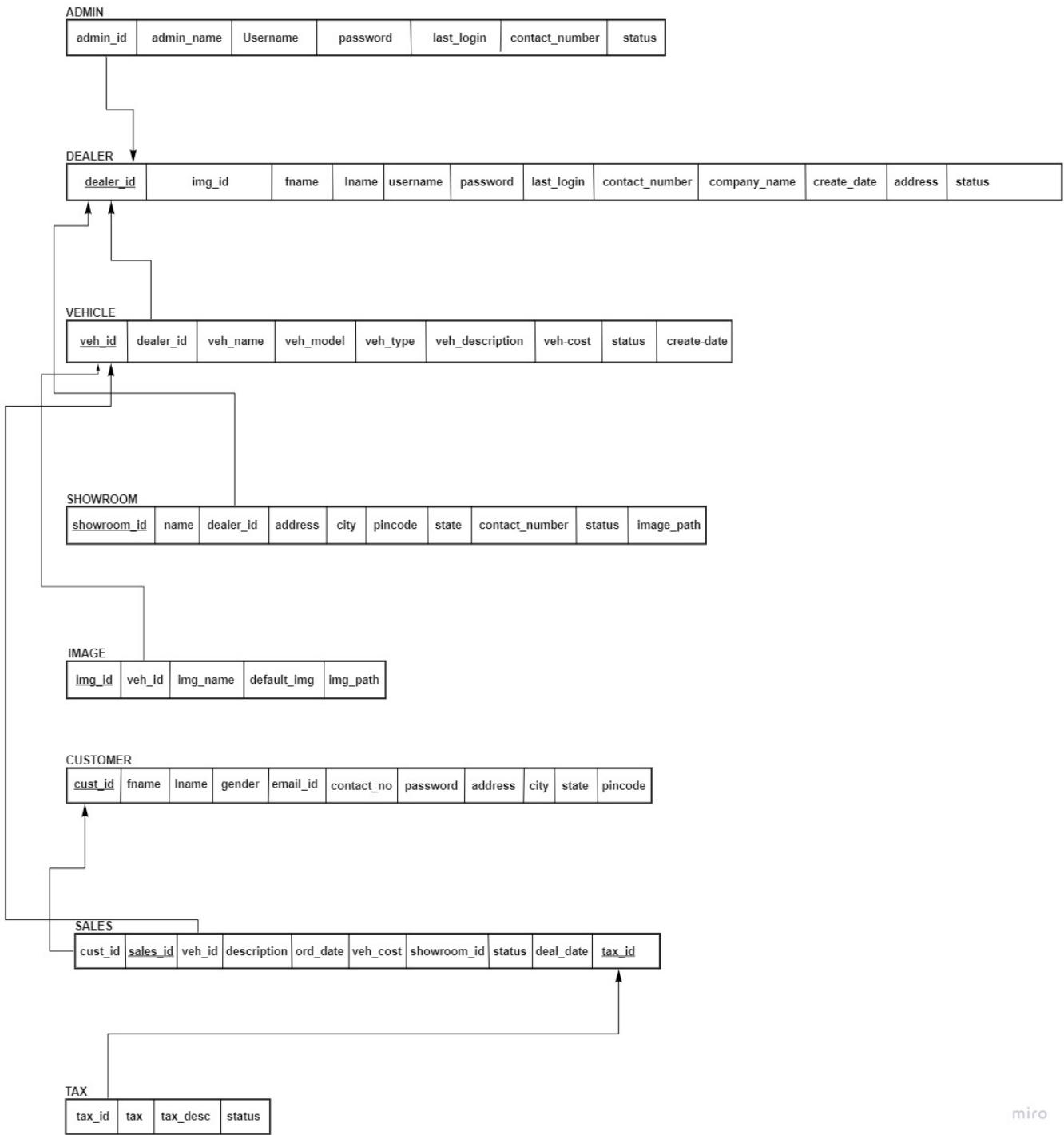
## **3.2 Database description:**

### **ER Diagram**



miro

## Schema Diagram



# Documentation

## 4. Backend

### 4.1 Models (Tables)

#### Dealer Model

miro

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

- dealer\_id

```
(variable) dealer_id: IntegerField => (
    primary_key: bool,
    unique: bool
)
```

```
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**.

- 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`

- [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 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`

- [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 committing 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.

- 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.

- 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`

- 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`.

- 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`.

- 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.

- [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

- [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`.

- `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 committing deletion on the referenced table as well. the `to_field` is set to `dealer_id`

- `vehicle_img`

Attribute definition:

```
(variable) vehicle_img: ImageField => (
  blank: Any
)
```

Source code:

```
vehicle_img = models.ImageField(  
    blank=False  
)
```

The `vehicle_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.

- 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.

- 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 vehicle type, its `max_length` is set `100` and `blank=False` so it cannot be left empty. The `choices` parameter is set to `VEHICLE_TYPE`

- [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`.

- 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`.

- 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 is `For Sale`.

- [created](#)

Attribute definition:

```
(variable) created: DateTimeField => (
  (parameter) auto_now_add: bool
)
```

Source code:

```
created = models.DateTimeField(
  auto_now_add=True
)
```

The `created` attribute 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

- 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`.

- `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`.

- `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`.

- 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 `choices` defined in a list of sets defined in `GENDER`.

- 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`.

- 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`.

- 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`.

- 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`.

- 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

- 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.

- [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.

- 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.

- 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.

- 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.

- 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.

- 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.

- 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.

- [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` attribute 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

- `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` attribute refers to the numeric id referenced in the Sale Model/Table.

- `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.

- 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 the description of the tax paid.

- 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.

## Snapshots

### 5.1 Frontend w/backend admin panel

127.0.0.1:8000

127.0.0.1:8000

Apps View Passwords (17) YouTube Google Online Courses - A... GitHub Programming articles Get AWS Certified S... Problemset - Codef... workouts megathread/games... Reading list

Buzzcars Home Vehicles Services About Us Contact Us Login SignUp

# FIND YOUR CAR



Explore Cars

127.0.0.1:8000

127.0.0.1:8000

Apps View Passwords (17) YouTube Google Online Courses - A... GitHub Programming articles Get AWS Certified S... Problemset - Codef... workouts megathread/games... Reading list

Buzzcars Home Vehicles Services About Us Contact Us Login SignUp

 150+ Branches

 4770+ Cars Sold

 320+ Happy Clients

 1500+ News Cars

## Popular Vehicles



### New Model

Price: **Rs.30,62,000/-**

New • 2021 • Automatic • Petrol • 183mph

Book A Quote

The screenshot shows a web browser window for the 'Buzzcars' website at 127.0.0.1:8000. The header includes a logo, navigation links for Home, Vehicles, Services, About Us, Contact Us, and buttons for Login and SignUp. The main section features a large heading 'Our Services' above six service cards. Each card has an icon, a title, a short description, and a 'Read More' button. The services listed are: Car Selling (car icon), Parts Repair (wrench icon), Car Insurance (car with exclamation icon), Battery Replacement (battery icon), Oil Change (gas pump icon), and 24/7 Support (headset icon). The descriptions for all services are identical placeholder text: 'Lorem, Ipsum Dolor Sit Amet Consectetur Adipisicing Elit. Corporis, Nisi.'.

The screenshot shows a web browser window for the URL 127.0.0.1:8000. The page title is "Buzzcars". The main content area features a large heading "Contact Us" above a map of Mumbai's northern suburbs. The map highlights areas like Jogeshwari West, Goregaon, Oshiwara, Andheri West, and Thane. Key landmarks marked on the map include Umiya Mata Mandir, SRV Hospital, Movietime: The Hub Mall, PVR Goregaon West, City Centre Mall, Carnival Cinemas, Imagica Tickets, Infiniti Mall, 1BHK, Oves, and various temples and railway stations. Below the map is a "Get In Touch" form with fields for First Name, Last Name, Gender, Email, Contact, City, and a message input field.

Buzzcars

Home Vehicles Services About Us Contact Us

Login

SignUp

# Contact Us

Jogeshwari West  
Mumbai, Maharashtra 400047

Directions

View larger map

H NAGAR

Umiya Mata Mandir

GOREGAON

SRV Hospital

Movietime: The Hub Mall

PVR Goregaon West

Carnival Cinemas

Imagica Tickets

Infiniti Mall

1BHK

Oves

okhandwala Market Road

SS NAGAR

ANDHERI - WEST

Jogeshwari Caves

Matoshri N. Thackeray

Kokan Hospital

## Get In Touch

First Name:

Last Name:

Gender:

-----

Email:

Contact:

City:

City

127.0.0.1:8000/vehicle/ 127.0.0.1:8000/vehicle/

Apps View Passwords (17) YouTube Google Online Courses - A... GitHub Programming articles Get AWS Certified S... Problemset - Codef... workouts megathread/games... Reading list

## Buzzcars

Home Vehicles Services About Us Contact Us



**Mercedes SLS AMG**  
Coupe For Sale

The Mercedes-Benz SLS AMG (C197 / R197) Is A Front Mid-Engine, 2-Seater, Limited Production Car.

Rs.2000000

[Book A Quote](#)



**BMW M3 (1990)**  
Sedan For Sale

The BMW M3 Is A High-Performance Version Of The BMW 3 Series, Developed By BMW's In-House Motorsport.

Rs.1600000

[Book A Quote](#)



**KIA Sportage**  
SUV For Sale

Kia Sportage Is A 5 Seater Suv. The Kia Sportage Is Expected To Launch In India In April 2022.

Rs.2500000

[Book A Quote](#)



**Lotus Emira**  
Crossover For Sale

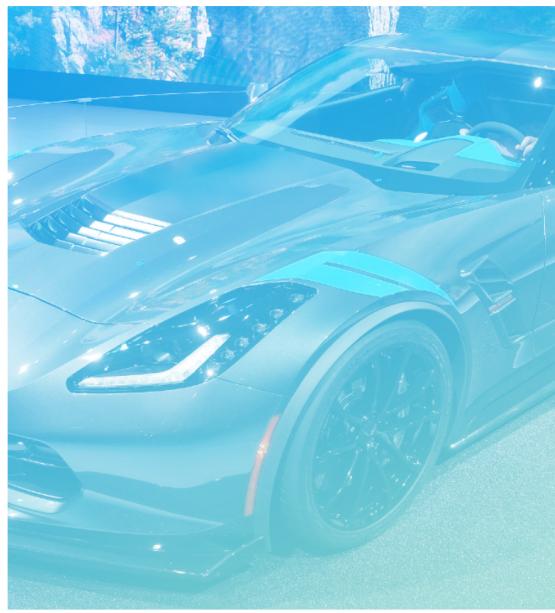
The Lotus Emira Is A Sports Car Manufactured By British Automobile Manufacturer Lotus Cars..

Rs.1500000000000

[Book A Quote](#)

127.0.0.1:8000/login/ 127.0.0.1:8000/login/

Apps View Passwords (17) YouTube Google Online Courses - A... GitHub Programming articles Get AWS Certified S... Problemset - Codef... workouts megathread/games... Reading list



## LOGIN

Username

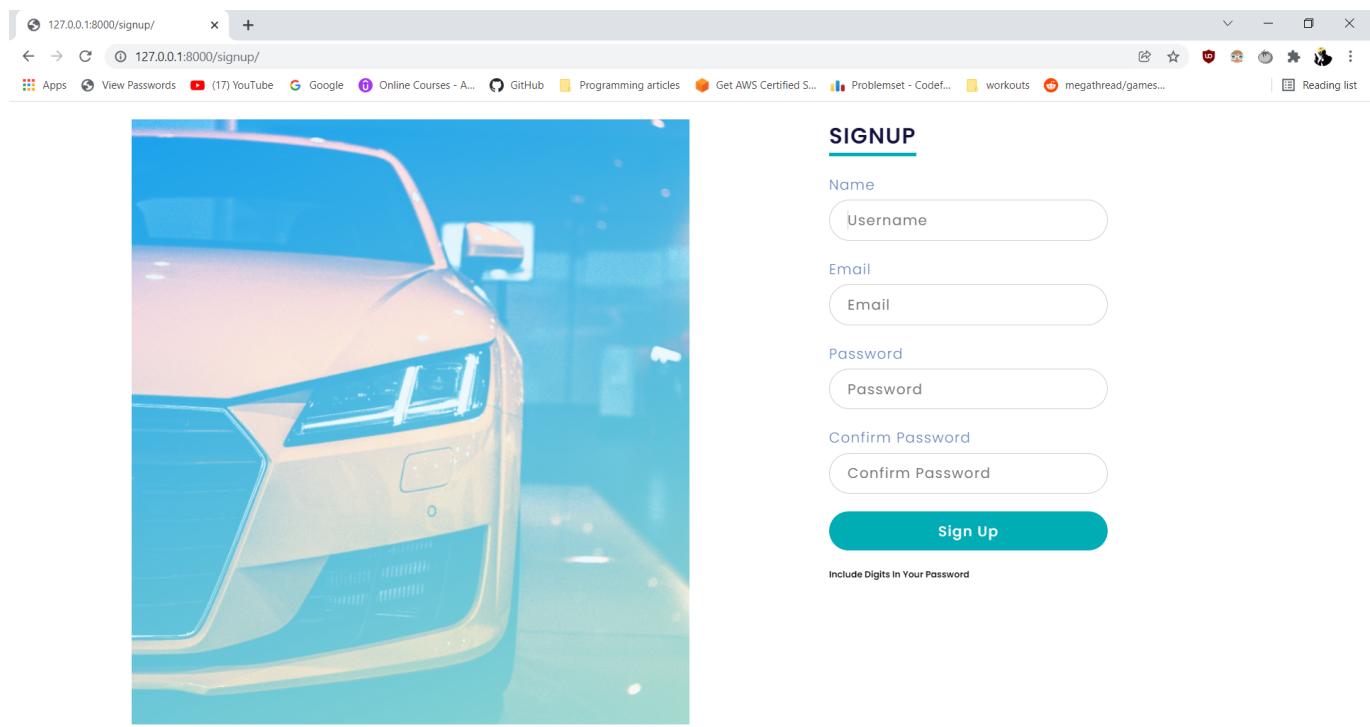
Password

Remember Me

[Sign In](#)

Or Login With

[Google](#) [Facebook](#)



A screenshot of the Django administration site. The top navigation bar shows 'Site administration | Django site'. The main content area is titled 'Django administration' and 'Site administration'. It contains two sections: 'AUTHENTICATION AND AUTHORIZATION' (Groups, Users) and 'SRC' (Customers, Dealers, Sales, Taxes, Vehicles). On the right, there are two panels: 'Recent actions' (listing recent vehicle and customer additions) and 'My actions' (listing recent vehicle and customer additions).



# Source code

File Edit Selection View Go Run Terminal Help

models.py - Buzzcars - Visual Studio Code

BUZZCARS

- src
  - \_\_pycache\_\_
  - migrations
  - \_init\_.py
  - admin.py
  - apps.py
  - forms.py
  - models.py
  - tests.py
  - urls.py
  - views.py
- static
- templates
- db.sqlite3
- manage.py
- env
- Synopsis.md

```
from email.headerregistry import Address
from re import A
from django.db import models
from django.contrib.auth.models import User
# Create your models here.

STATUS_CHOICE = [ ('Active', 'Active'), ('Inactive', 'Inactive')]

class Dealer(models.Model):
    dealer_id = models.IntegerField(primary_key=True, unique=True)
    dealer_img = models.ImageField(blank=False)
    dealer_name = models.CharField(max_length=100, blank=False, null=False)
    dealer_username = models.ForeignKey(User, on_delete=models.CASCADE, related_name='d_username')
    last_login = models.DateTimeField(auto_now=True)
    contact = models.IntegerField(null=False, blank=False)
    company_name = models.CharField(max_length=100, null=False, blank=False)
    address = models.TextField(max_length=300, blank=False, null=False)
    created = models.DateTimeField(auto_created=True)
    status = models.CharField(max_length=20, choices=STATUS_CHOICE, default='Active')

    def __str__(self):
        return str(self.dealer_id)

VEHICLE_TYPE = [
    ("Hatchback", "Hatchback"),
    ("Sedan", "Sedan"),
    ("MPV", "MPV"),
    ("SUV", "SUV"),
    ("Crossover", "Crossover"),
    ("Coupe", "Coupe"),
    ("Convertible", "Convertible")
]

VEH_STATUS = [
    ("For Sale", "For Sale"),
    ("Sold", "Sold")
]

class Vehicle(models.Model):
    vehicle_id = models.IntegerField(primary_key=True, unique=True)
    dealer_id = models.ForeignKey(Dealer, to_field='dealer_id', on_delete=models.CASCADE)
    vehicle_img = models.ImageField(blank=False)
    name = models.CharField(max_length=100, blank=False)
    type = models.CharField(max_length=100, choices=VEHICLE_TYPE, blank=False)
    description = models.TextField(max_length=100, blank=False)
    cost = models.IntegerField(blank=False)
    status = models.CharField(max_length=20, choices=VEH_STATUS, blank=False, default='For Sale')
    created = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.name

GENDER = [
    ("Male", "Male"),
    ("Female", "Female"),
    ("Other", "Other")
]

class Customer(models.Model):
    customer_id = models.IntegerField(primary_key=True, unique=True)
```

Ln 48, Col 98 Spaces: 4 UTF-8 CRLF Python ⚡ Prettier ⌂ ENG 8:37 PM IN 1/19/2022 (3)

File Edit Selection View Go Run Terminal Help

models.py - Buzzcars - Visual Studio Code

BUZZCARS

- src
  - \_\_pycache\_\_
  - migrations
  - \_init\_.py
  - admin.py
  - apps.py
  - forms.py
  - models.py
  - tests.py
  - urls.py
  - views.py
- static
- templates
- db.sqlite3
- manage.py
- env
- Synopsis.md

```
VEHICLE_TYPE = [
    ("Hatchback", "Hatchback"),
    ("Sedan", "Sedan"),
    ("MPV", "MPV"),
    ("SUV", "SUV"),
    ("Crossover", "Crossover"),
    ("Coupe", "Coupe"),
    ("Convertible", "Convertible")
]

VEH_STATUS = [
    ("For Sale", "For Sale"),
    ("Sold", "Sold")
]

class Vehicle(models.Model):
    vehicle_id = models.IntegerField(primary_key=True, unique=True)
    dealer_id = models.ForeignKey(Dealer, to_field='dealer_id', on_delete=models.CASCADE)
    vehicle_img = models.ImageField(blank=False)
    name = models.CharField(max_length=100, blank=False)
    type = models.CharField(max_length=100, choices=VEHICLE_TYPE, blank=False)
    description = models.TextField(max_length=100, blank=False)
    cost = models.IntegerField(blank=False)
    status = models.CharField(max_length=20, choices=VEH_STATUS, blank=False, default='For Sale')
    created = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.name

GENDER = [
    ("Male", "Male"),
    ("Female", "Female"),
    ("Other", "Other")
]

class Customer(models.Model):
    customer_id = models.IntegerField(primary_key=True, unique=True)
```

Ln 48, Col 98 Spaces: 4 UTF-8 CRLF Python ⚡ Prettier ⌂ ENG 8:37 PM IN 1/19/2022 (3)

File Edit Selection View Go Run Terminal Help

models.py - Buzzcars - Visual Studio Code

EXPLORER BUZZCARS Buzzcars media src \_pycache\_ migrations \_\_init\_\_.py admin.py apps.py forms.py models.py tests.py urls.py views.py static templates db.sqlite3 manage.py env Synopsis.md

```
Buzzcars > src > models.py > Vehicle
    ('other', 'other')
]
]

class Customer(models.Model):
    customer_id = models.IntegerField(primary_key=True, unique=True)
    first_name = models.CharField(max_length=100, blank=False)
    last_name = models.CharField(max_length=100, blank=False)
    gender = models.CharField(max_length=100, blank=False, choices=GENDER)
    email = models.CharField(max_length=100, blank=False)
    contact = models.IntegerField(blank=False)
    city = models.CharField(max_length=100, blank=False)
    state = models.CharField(max_length=100, blank=False)
    address = models.TextField(max_length=100, blank=False)

    def __str__(self):
        return self.first_name + " " + self.last_name

SALE_STATUS = [
    ("Sold", "Sold"),
    ("On hold", "On hold"),
    ("Rejected", "Rejected")
]

class Sale(models.Model):
    customer_id = models.ForeignKey(Customer, to_field='customer_id', on_delete=models.CASCADE)
    sale_id = models.IntegerField(primary_key=True, unique=True)
    vehicle_id = models.ForeignKey(Vehicle, to_field='vehicle_id', on_delete=models.CASCADE)
    description = models.TextField(max_length=100, blank=False)
    order_date = models.DateTimeField(auto_created=True)
    cost = models.IntegerField(blank=False)
    deal_date = models.DateTimeField(auto_created=True)
    status = models.CharField(max_length=20, blank=False, choices=SALE_STATUS)
    tax_id = models.IntegerField(unique=True, blank=False)

    def __str__(self):
        return str(self.vehicle_id)

TAX_STATUS = [
```

Python 3.9.6 64-bit ('env': venv) 0 0 0 Rahul Live Share

Ln 48, Col 98 Spaces: 4 UTF-8 CRLF Python Prettier ENG 8:37 PM IN 1/19/2022

File Edit Selection View Go Run Terminal Help

models.py - Buzzcars - Visual Studio Code

EXPLORER BUZZCARS Buzzcars media src \_pycache\_ migrations \_\_init\_\_.py admin.py apps.py forms.py models.py tests.py urls.py views.py static templates db.sqlite3 manage.py env Synopsis.md

```
Buzzcars > src > models.py > Vehicle
    ('other', 'other')
]
]

class Customer(models.Model):
    customer_id = models.ForeignKey(Customer, to_field='customer_id', on_delete=models.CASCADE)
    sale_id = models.IntegerField(primary_key=True, unique=True)
    vehicle_id = models.ForeignKey(Vehicle, to_field='vehicle_id', on_delete=models.CASCADE)
    description = models.TextField(max_length=100, blank=False)
    order_date = models.DateTimeField(auto_created=True)
    cost = models.IntegerField(blank=False)
    deal_date = models.DateTimeField(auto_created=True)
    status = models.CharField(max_length=20, blank=False, choices=SALE_STATUS)
    tax_id = models.IntegerField(unique=True, blank=False)

    def __str__(self):
        return str(self.vehicle_id)

TAX_STATUS = [
    ("Approved", "Approved"),
    ("Pending", "Pending"),
    ("Rejected", "Rejected")
]

class Tax(models.Model):
    tax_id = models.ForeignKey(Sale, to_field='tax_id', on_delete=models.CASCADE)
    tax = models.IntegerField(blank=False)
    description = models.TextField(max_length=100, blank=False)
    status = models.CharField(max_length=20, blank=False, choices=TAX_STATUS)

    def __str__(self):
        return str(self.tax_id)
```

Python 3.9.6 64-bit ('env': venv) 0 0 0 Rahul Live Share

Ln 48, Col 98 Spaces: 4 UTF-8 CRLF Python Prettier ENG 8:37 PM IN 1/19/2022

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** forms.py - Buzzcars - Visual Studio Code
- Left Sidebar (EXPLORER):** BUZZCARS folder containing files: \_\_pycache\_\_, migrations, \_\_init\_\_.py, admin.py, apps.py, forms.py, models.py, tests.py, urls.py, views.py, static, templates, db.sqlite3, manage.py, env, and Synopsis.md.
- Central Area:** Code editor showing the contents of forms.py. The code defines a ContactForm class that inherits from ModelForm. It overrides the \_\_init\_\_ method to update widget attributes for first\_name, last\_name, gender, and email fields. The \_\_init\_\_ method calls super().\_\_init\_\_(args, \*\*kwargs). The code uses Python's f-string syntax for placeholder values.
- Bottom Status Bar:** Python 3.9.6 64-bit (env: venv), Rahul, Live Share, Ln 63, Col 33, Spaces: 5, UTF-8, CRLF, Python, Prettier, ENG IN 1/19/2022, 8:38 PM.

File Edit Selection View Go Run Terminal Help

forms.py - Buzzcars - Visual Studio Code

EXPLORER BUZZCARS

forms.py > Buzzcars > src > forms.py > ContactForm > \_\_init\_\_

```
74     'required':''  
75     })  
76  
77     self.fields['email'].widget.attrs.update({  
78         'type': 'text',  
79         'name': 'email',  
80         'placeholder': 'Email',  
81         'class': 'box',  
82         'required':''  
83     })  
84  
85     self.fields['contact'].widget.attrs.update({  
86         'type': 'text',  
87         'name': 'contact',  
88         'placeholder': 'Contact',  
89         'class': 'box',  
90         'required':''  
91     })  
92  
93     self.fields['city'].widget.attrs.update({  
94         'type': 'text',  
95         'name': 'city',  
96         'placeholder': 'City',  
97         'class': 'box',  
98         'required':''  
99     })  
100  
101     self.fields['state'].widget.attrs.update({  
102         'type': 'text',  
103         'name': 'state',  
104         'placeholder': 'State',  
105         'class': 'box',  
106         'required':''  
107     })  
108  
109     self.fields['address'].widget.attrs.update({  
110         'type': 'text',  
111         'name': 'address'  
112     })  
113  
114 class Meta:  
115     model = Customer  
116     fields = ['first_name', 'last_name', 'gender', 'email', 'contact', 'city', 'state', 'address']
```

Python 3.9.6 64-bit ('env': venv) 0 0 0 Rahul Live Share

Ln 63, Col 33 Spaces: 5 UTF-8 CRLF Python Prettier ENG 8:38 PM IN 1/19/2022

File Edit Selection View Go Run Terminal Help

forms.py - Buzzcars - Visual Studio Code

EXPLORER BUZZCARS

forms.py > Buzzcars > src > forms.py > ContactForm > \_\_init\_\_

```
97     'class': 'box',  
98     'required':''  
99     })  
100  
101     self.fields['state'].widget.attrs.update({  
102         'type': 'text',  
103         'name': 'state',  
104         'placeholder': 'State',  
105         'class': 'box',  
106         'required':''  
107     })  
108  
109     self.fields['address'].widget.attrs.update({  
110         'type': 'text',  
111         'name': 'address',  
112         'placeholder': 'Address',  
113         'class': 'box',  
114         'required':''  
115     })  
116  
117  
118 class Meta:  
119     model = Customer  
120     fields = ['first_name', 'last_name', 'gender', 'email', 'contact', 'city', 'state', 'address']
```

Python 3.9.6 64-bit ('env': venv) 0 0 0 Rahul Live Share

Ln 63, Col 33 Spaces: 5 UTF-8 CRLF Python Prettier ENG 8:38 PM IN 1/19/2022

File Edit Selection View Go Run Terminal Help

admin.py - Buzzcars - Visual Studio Code

EXPLORER BUZZCARS admin.py

```
BUZZCARS
  Buzzcars
  media
  src
    _pycache_
    migrations
    __init__.py
    admin.py
    apps.py
    forms.py
    models.py
    tests.py
    urls.py
    views.py
  static
  templates
  db.sqlite3
  manage.py
  env
  Synopsis.md
```

admin.py

```
from django.contrib import admin
from .models import Dealer, Vehicle, Customer, Sale, Tax

# Register your models here.
admin.site.register(Dealer)
admin.site.register(Vehicle)
admin.site.register(Customer)
admin.site.register(Sale)
admin.site.register(Tax)
```

OUTLINE

Python 3.9.6 64-bit ('env': venv) 0 0 0 Rahul Live Share

Ln 9, Col 25 Spaces: 4 UTF-8 CRLF Python Prettier ENG 8:38 PM IN 1/19/2022

File Edit Selection View Go Run Terminal Help

urls.py - Buzzcars - Visual Studio Code

EXPLORER BUZZCARS urls.py

```
BUZZCARS
  Buzzcars
  media
  src
    _pycache_
    migrations
    __init__.py
    admin.py
    apps.py
    forms.py
    models.py
    tests.py
    urls.py
    views.py
  static
  templates
  db.sqlite3
  manage.py
  env
  Synopsis.md
```

urls.py

```
from django.contrib import admin
from django.urls import path, include
from . import views
from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [
    path('', views.Home, name='home'),
    path('about/', views.About, name='about'),
    path('login/', views.loginPage, name='login'),
    path('signup/', views.Signup, name='signup'),
    path('logout/', views.logoutUser, name='logout'),
    path('vehicle/', views.vehiclePage, name='vehicle'),
]
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

OUTLINE

Python 3.9.6 64-bit ('env': venv) 0 0 0 Rahul Live Share

Ln 8, Col 36 (4 selected) Spaces: 4 UTF-8 CRLF Python Prettier ENG 8:38 PM IN 1/19/2022

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure for "BUZZCARS". The "src" folder contains files like `__init__.py`, `admin.py`, `apps.py`, `forms.py`, `models.py`, `tests.py`, and `urls.py`. The current file being edited is `views.py`.
- Code Editor (Center):** Displays the content of `views.py`. The code defines two functions: `def Home(request)` and `def loginPage(request)`. The `Home` function handles POST requests by saving a `ContactForm` if it's valid. The `loginPage` function handles POST requests by trying to authenticate a user with the provided credentials.
- Status Bar (Bottom):** Shows the Python version (3.9.6), environment (env), file path (views.py), line count (Ln 21), column count (Col 1), and other system information.

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer Sidebar:** Shows the project structure under BUZZCARS:
  - BUZZCARS
  - Buzzcars
  - Buzzcars
  - media
  - src
    - \_\_pycache\_\_
    - migrations
    - \_\_init\_\_.py
    - admin.py
    - apps.py
    - forms.py
    - models.py
    - tests.py
    - urls.py
    - views.py
  - static
  - templates
  - db.sqlite3
  - manage.py
  - env
  - Synopsis.md
- Editor Area:** The file `views.py` is open, showing Python code for a web application. The code handles user authentication (login, logout, signup) and vehicle management. It uses Django's `render` function to return HTML templates and `messages.error` to handle user errors.
- Bottom Status Bar:** Python 3.9.6 64-bit (env: venv), Rahul, Live Share, Ln 21, Col 1, Spaces: 4, UTF-8, CRLF, Python, Prettier.

File Edit Selection View Go Run Terminal Help

about.html - Buzzcars - Visual Studio Code

EXPLORER BUZZCARS

- Buzzcars
- index.html
- login.html
- signup.html
- vehicles.html
- db.sqlite3
- manage.py
- env
- Synopsis.md

about.html

```
1  <% load static %>
2  <!DOCTYPE html>
3  <html lang="en">
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8
9      <link rel="stylesheet" href="https://unpkg.com/swiper@7/swiper-bundle.min.css" />
10
11     <!-- font awesome cdn Link -->
12     <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css">
13
14     <!-- custom css file Link -->
15     <link rel="stylesheet" href="<% static 'css/style.css' %>">
16
17 </head>
18 <body>
19
20     <header class="header">
21         <div id="menu-btn" class="fas fa-bars"></div>
22
23         <a href="#" class="logo"> maxwheels </a>
24
25         <nav class="navbar">
26             <a href="<% url 'home' %>">home</a>
27             <a href="<% url 'about' %>">About Us</a>
28             <a href="<% url 'vehicle' %>">Vehicles</a>
29         </nav>
30     </header>
31
32     <div class="about-section">
33         <div class="about-containers">
34             <div class="content-section">
35                 <div class="title">
36                     <h1>About Us</h1>
37                     <div class="border"></div>
38             </div>
39         </div>
40     </div>
41
42     <div class="content">
43         <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Corrupti quod molestiae quae accusantium fuga? Consectetur aperi
44
45         </div>
46         <div class="skills-title">
47             <h2>We Serve!</h2>
48         </div>
49         <div class="skills">
50             <span>Quality services</span>
51             <span>Premium Cars</span>
52             <span>Quality Parts</span>
53         </div>
54         <div class="image-section">
55             
56         </div>
57     </div>
58 </body>
59 </html>
```

Python 3.9.6 64-bit ('env': venv) 0 0 0 Rahul Live Share

Ln 21, Col 1 Spaces: 4 UTF-8 CRLF HTML Go Live Prettier ENG 8:39 PM IN 1/19/2022

File Edit Selection View Go Run Terminal Help

about.html - Buzzcars - Visual Studio Code

EXPLORER BUZZCARS

- Buzzcars
- index.html
- login.html
- signup.html
- vehicles.html
- db.sqlite3
- manage.py
- env
- Synopsis.md

about.html

```
1  </header>
2  <div class="about-section">
3      <div class="about-containers">
4          <div class="content-section">
5              <div class="title">
6                  <h1>About Us</h1>
7                  <div class="border"></div>
8              </div>
9              <div class="content">
10                 <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Corrupti quod molestiae quae accusantium fuga? Consectetur aperi
11
12                 </div>
13                 <div class="skills-title">
14                     <h2>We Serve!</h2>
15                 </div>
16                 <div class="skills">
17                     <span>Quality services</span>
18                     <span>Premium Cars</span>
19                     <span>Quality Parts</span>
20                 </div>
21             </div>
22         </div>
23     </div>
24 </body>
25 </html>
```

Python 3.9.6 64-bit ('env': venv) 0 0 0 Rahul Live Share

Ln 21, Col 1 Spaces: 4 UTF-8 CRLF HTML Go Live Prettier ENG 8:39 PM IN 1/19/2022

index.html - Buzzcars - Visual Studio Code

```
BUZZCARS > templates > index.html > ...
index.html
1 1% load static %
2 <!DOCTYPE html>
3 <html lang="en">
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8
9   <link rel="stylesheet" href="https://unpkg.com/swiper@7/swiper-bundle.min.css" />
10
11  <!-- font awesome cdn Link -->
12  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css">
13
14  <!-- custom css file Link -->
15  <link rel="stylesheet" href="{% static 'css/style.css' %}">
16
17 </head>
18 <body>
19
20 <header class="header">
21
22   <div id="menu-btn" class="fas fa-bars"></div>
23
24   <a href="#" class="logo"> <span>Buzz</span>cars </a>
25
26   <nav class="navbar">
27     <a href="{% url 'home' %}">home</a>
28     <a href="{% url 'vehicle' %}">vehicles</a>
29     <a href="#services">services</a>
30     <a href="{% url 'about' %}">About Us</a>
31     <a href="#contact">Contact Us</a>
32   </nav>
33
34   <div id="login-btn">
35     <div class="user-container">
36
37       1% if request.user.is_authenticated %>
38
39       <p class="user"><span>Hello {{request.user}} !</span></p>
40       <button class="btn"><a href="{% url 'logout' %}"><span>Logout</span></a></button>
41
42       &% else %
43       <button class="btn"><a href="{% url 'login' %}"><span>Login</span></a></button>
44
45       &% endif %
46     </div>
47
48     <i class="far fa-user"></i>
49   </div>
50
51   <div id="signup-btn">
52     <button class="btn"><a href="{% url 'signup' %}"><span>SignUp</span></a></button>
53     <i class="far fa-user"></i>
54   </div>
55
56 </header>
57
58 <section class="home" id="home">
59   <h3 data-speed="-2" class="home-parallax">find your car</h3>
60
61   
62
63   <a data-speed="7" href="{% url 'vehicle' %}" class="btn home-parallax">explore cars</a>
64
65 </section>
66
67 <section class="icons-container">
68   <div class="icons">
69     <i class="fas fa-home"></i>
70     <div class="content">
71       <h3>150+</h3>
72       <p>branches</p>
73     </div>
74   </div>
75
76 </section>
```

index.html - Buzzcars - Visual Studio Code

```
BUZZCARS > templates > index.html > ...
index.html
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure under "BUZZCARS". The "index.html" file is selected and highlighted.
- Editor (Center):** Displays the content of the "index.html" file. The code includes sections for icons, content, and a vehicle slider.
- Terminal (Bottom Left):** Shows the command "Rahul" entered.
- Status Bar (Bottom):** Provides information about the Python environment ("Python 3.9.6 64-bit (Env: venv)"), the current file ("index.html - Buzzcars - Visual Studio Code"), and the terminal session ("Rahul").
- Activity Bar (Bottom Right):** Includes icons for Share, Live Share, Go Live, and Prettier.

```
<div class="icons">
    <i class="fas fa-car"></i>
    <div class="content">
        <h3>4770+</h3>
        <p>cars sold</p>
    </div>
</div>

<div class="icons">
    <i class="fas fa-users"></i>
    <div class="content">
        <h3>20+</h3>
        <p>happy clients</p>
    </div>
</div>

<div class="icons">
    <i class="fas fa-car"></i>
    <div class="content">
        <h3>1500+</h3>
        <p>news cars</p>
    </div>
</div>

</section>

<section class="vehicles" id="vehicles">

    <h1 class="heading"> popular <span>vehicles</span> </h1>

    <div class="swiper vehicles-slider">
        <div class="swiper-wrapper">
            <div class="swiper-slide box">
                
                <div class="content">
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer:** BUZZCARS folder containing: Buzzcars, media, src, static, templates (about.html, index.html, login.html, signup.html, vehicles.html), db.sqlite3, env, Synopsis.md.
- Editor:** The index.html file is open, displaying three car models with their details and a "Book a Quote" button.
- Bottom Status Bar:** Python 3.9.6 64-bit (env: venv), Rahul, Live Share, L1, Col 12, Spaces: 4, UTF-8, CRLF, HTML, Go Live, IN 1/19/2022, ENG 8:39 PM.

```

<div class="content">
  <h3>new model</h3>
  <div class="price"> <span>price : </span> Rs.28,62,000/- </div>
  <p>
    new
    <span class="fas fa-circle"></span> 2021
    <span class="fas fa-circle"></span> automatic
    <span class="fas fa-circle"></span> petrol
    <span class="fas fa-circle"></span> 183mph
  </p>
  <a href="#contact" class="btn">Book a Quote</a>
</div>

<div class="swiper-slide box">
  
  <div class="content">
    <h3>new model</h3>
    <div class="price"> <span>price : </span> Rs.30,62,000/- </div>
    <p>
      new
      <span class="fas fa-circle"></span> 2021
      <span class="fas fa-circle"></span> automatic
      <span class="fas fa-circle"></span> petrol
      <span class="fas fa-circle"></span> 183mph
    </p>
    <a href="#contact" class="btn">Book a Quote</a>
  </div>
</div>

<div class="swiper-slide box">
  
  <div class="content">
    <h3>new model</h3>
    <div class="price"> <span>price : </span> Rs.34,62,000/- </div>
    <p>
      new
    </p>
  </div>
</div>
```

File Edit Selection View Go Run Terminal Help index.html - Buzzcars - Visual Studio Code

EXPLORER BUZZCARS ... index.html x

Buzzcars > templates > index.html > ...

```
180     <span class="fas fa-circle">2021
181     <span class="fas fa-circle">automatic
182     <span class="fas fa-circle">petrol
183     <span class="fas fa-circle">183mph
184   </p>
185   <a href="#contact" class="btn">Book a Quote</a>
186 </div>
187 </div>

188 <div class="swiper-slide box">
189   
190   <div class="content">
191     <h3>new model</h3>
192     <div class="price"> <span>price : </span> Rs.29,62,000/- </div>
193     <p>
194       new
195       <span class="fas fa-circle">2021
196       <span class="fas fa-circle">automatic
197       <span class="fas fa-circle">petrol
198       <span class="fas fa-circle">183mph
199     </p>
200     <a href="#contact" class="btn">Book a Quote</a>
201   </div>
202 </div>

203 <div class="swiper-pagination"></div>
204 </div>
205 </section>
206
207 <section class="services" id="services">
208   <h1 class="heading"> our <span>services</span> </h1>
209   <div class="box-container">
210     <div class="swiper-pagination"></div>
211   </section>
212
213 <section class="services" id="services">
214   <h1 class="heading"> our <span>services</span> </h1>
215   <div class="box-container">
216     <div class="box">
217       <i class="fas fa-car"></i>
218       <h3>car selling</h3>
219       <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Corporis, nisi.</p>
220       <a href="#" class="btn">read more</a>
221     </div>
222     <div class="box">
223       <i class="fas fa-tools"></i>
224       <h3>parts repair</h3>
225       <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Corporis, nisi.</p>
226       <a href="#" class="btn">read more</a>
227     </div>
228     <div class="box">
229       <i class="fas fa-car-crash"></i>
230       <h3>car insurance</h3>
231       <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Corporis, nisi.</p>
232       <a href="#" class="btn">read more</a>
233     </div>
234     <div class="box">
235       <i class="fas fa-car-battery"></i>
236       <h3>battery replacement</h3>
237       <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Corporis, nisi.</p>
238       <a href="#" class="btn">read more</a>
239     </div>
240   </div>
241 </section>
242
243 <div class="box">
244   <h3>Our Services</h3>
245   <ul>
246     <li>Car Selling</li>
247     <li>Parts Repair</li>
248     <li>Car Insurance</li>
249     <li>Battery Replacement</li>
250   </ul>
251 </div>
```

Ln 1, Col 12 Spaces: 4 UTF-8 CRLF HTML ⚡ Go Live ✨ Prettier ⌂ ENG 8:39 PM IN 1/19/2022

File Edit Selection View Go Run Terminal Help index.html - Buzzcars - Visual Studio Code

EXPLORER BUZZCARS ... index.html x

Buzzcars > templates > index.html > ...

```
207   <div class="swiper-pagination"></div>
208 </div>
209 </section>
210
211 <section class="services" id="services">
212   <h1 class="heading"> our <span>services</span> </h1>
213   <div class="box-container">
214     <div class="box">
215       <i class="fas fa-car"></i>
216       <h3>car selling</h3>
217       <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Corporis, nisi.</p>
218       <a href="#" class="btn">read more</a>
219     </div>
220     <div class="box">
221       <i class="fas fa-tools"></i>
222       <h3>parts repair</h3>
223       <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Corporis, nisi.</p>
224       <a href="#" class="btn">read more</a>
225     </div>
226     <div class="box">
227       <i class="fas fa-car-crash"></i>
228       <h3>car insurance</h3>
229       <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Corporis, nisi.</p>
230       <a href="#" class="btn">read more</a>
231     </div>
232     <div class="box">
233       <i class="fas fa-car-battery"></i>
234       <h3>battery replacement</h3>
235       <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Corporis, nisi.</p>
236       <a href="#" class="btn">read more</a>
237     </div>
238   </div>
239 </section>
240
241 <div class="box">
242   <h3>Our Services</h3>
243   <ul>
244     <li>Car Selling</li>
245     <li>Parts Repair</li>
246     <li>Car Insurance</li>
247     <li>Battery Replacement</li>
248   </ul>
249 </div>
```

Ln 1, Col 12 Spaces: 4 UTF-8 CRLF HTML ⚡ Go Live ✨ Prettier ⌂ ENG 8:39 PM IN 1/19/2022

File Edit Selection View Go Run Terminal Help

index.html

```
Buzzcars > templates > index.html ...  
263 </section>  
264  
265  
266 <section class="newsletter">  
267  
268 <h3>subscribe for latest updates</h3>  
269 <p>Lorem ipsum dolor sit amet consectetur adipisicing elit, voluptatum, suscipit.</p>  
270  
271 <form action="">  
272 <input type="email" placeholder="enter your email">  
273 <input type="submit" value="subscribe">  
274 </form>  
275  
276 </section>  
277  
278 <section class="contact" id="contact">  
279  
280 <h1 class="heading"><span>contact</span> us</h1>  
281  
282 <div class="row">  
283  
284 <iframe class="map" src="https://www.google.com/maps/embed?pb=1m18!1m12!1m3!1d30153.788252261566!2d72.82321484621745!3d19.141690214" ...  
285  
286 <form method="POST" action="">  
287 <h3>Get in touch</h3>  
288 <% csrf_token %>  
289 <!-- <input type="text" placeholder="your name" class="box">  
290 <input type="email" placeholder="your email" class="box">  
291 <input type="tel" placeholder="subject" class="box">  
292 <textarea placeholder="your message" class="box" cols="30" rows="10"></textarea>  
293 <input type="submit" value="send message" class="btn"> -->  
294 {{form}}  
295 <input type="submit" value="send message" class="btn">  
296 </form>  
297  
298 </div>  
299  
300 </section>
```

Ln 1, Col 12 Spaces: 4 UTF-8 CRLF HTML ⚡ Go Live ✨ Prettier 🔍 ENG IN 8:40 PM 1/19/2022

File Edit Selection View Go Run Terminal Help

index.html

```
Buzzcars > templates > index.html ...  
303 <div class="box-container">  
304  
305  
306  
307 <div class="box">  
308 <h3>our branches</h3>  
309 <a href="#">  india </a>  
310 <a href="#">  japan </a>  
311 <a href="#">  france </a>  
312 <a href="#">  russia </a>  
313 <a href="#">  USA </a>  
314 </div>  
315  
316 <div class="box">  
317 <h3>quick links</h3>  
318 <a href="{% url 'home' %}">  home </a>  
319 <a href="{% url 'vehicle' %}">  vehicles </a>  
320 <a href="{% url 'services' %}">  services </a>  
321 <a href="{% url 'about' %}">  About Us </a>  
322 <a href="#contact">  Contact Us </a>  
323 </div>  
324  
325 <div class="box">  
326 <h3>contact info</h3>  
327 <a href="#">  9843273637 </a>  
328 <a href="#">  9637536687 </a>  
329 <a href="#">  saurav123@gmail.com </a>  
330 <a href="#">  Bangalore, India - 560083 </a>  
331 </div>  
332  
333 <div class="box">  
334 <h3>contact info</h3>  
335 <a href="#">  facebook </a>  
336 <a href="#">  twitter </a>  
337 <a href="#">  instagram </a>  
338 <a href="#">  linkedin </a>  
339 <a href="#">  pinterest </a>  
340 </div>
```

Ln 1, Col 12 Spaces: 4 UTF-8 CRLF HTML ⚡ Go Live ✨ Prettier 🔍 ENG IN 8:40 PM 1/19/2022

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure under "BUZZCARS". The "index.html" file is selected.
- Editor (Center):** Displays the content of "index.html". The code includes HTML for contact info, social media links (Facebook, Twitter, Instagram, LinkedIn, Pinterest), and a copyright notice. It also includes script tags for Swiper.js and a local script.js file.
- Terminal (Bottom Left):** Shows the command "Live Share".
- Status Bar (Bottom):** Shows the Python version (3.9.6 64-bit), environment (venv), file count (0), author (Rahul), and other status indicators like Live Share, Linters, and Language Services.
- Activity Bar (Bottom Right):** Shows icons for Go Live, Prettier, and other extensions.

```
<!-- index.html -->
<div class="box">
    <h3>Contact Info</h3>
    <a href="#"> <i class="fas fa-phone"></i> 9843273637 </a>
    <a href="#"> <i class="fas fa-phone"></i> 9637536687 </a>
    <a href="#"> <i class="fas fa-envelope"></i> saurav123@gmail.com </a>
    <a href="#"> <i class="fas fa-map-marker-alt"></i> Bangalore, India - 560083 </a>
</div>
<div class="credit"> Front-end by Saurav Kumar | All rights reserved </div>
</section>
<script src="https://unpkg.com/swiper@7/swiper-bundle.min.js"></script>
<script src="{% static 'JS/script.js' %}"></script>
</body>
</html>
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** On the left, the "BUZZCARS" folder is expanded, showing files like `about.html`, `index.html`, `login.html`, `signup.html`, and `vehicles.html`.
- Editor:** The main editor area displays the `login.html` file content. The code is a combination of HTML and Jinja2 templating. It includes meta tags, links to CSS files (Swiper and Font Awesome), and a section for displaying messages.
- Terminal:** At the bottom, the terminal shows the command `Python 3.9.6 64-bit (env: venv)`.
- Status Bar:** The status bar at the bottom right shows the current file is `login.html`, the line number is `Ln 29, Col 54`, and the word count is `Spaces: 4`. It also shows file formats `UTF-8`, `CRLF`, `HTML`, and a "Go Live" button.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** On the left, it shows the project structure under "BUZZCARS". The "login.html" file is selected.
- Editor:** The main area displays the content of "login.html". The code includes HTML structure like forms, input fields, and buttons, along with some Python-like templating syntax (e.g., {% for message in messages %}).
- Terminal:** At the bottom, the terminal shows the command "Rahul" followed by a "Live Share" icon.
- Status Bar:** The status bar at the bottom right indicates the file is "login.html - Buzzcars - Visual Studio Code", with "Ln 29, Col 54" and other file statistics.

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer:** BUZZCARS folder containing files like Buzzcars, Buzzcars, media, src, static, templates (about.html, index.html, login.html, signup.html), vehicles.html, db.sqlite3, manage.py, env, and Synopsis.md.
- Editor:** The 'signup.html' file is open in the editor. The code is a JSP template for a login page. It includes meta tags for charset, http-equiv, and viewport, as well as links to CSS files from Unpkg and Cloudflare. The main content section contains an image of a sign-up page, a form box with a 'SIGNUP' button, and a messages list. A csrf token is included in the form.
- Terminal:** Shows the command 'Python 3.9.6 64-bit (env: venv)'. Other tabs in the terminal include 'Rahul' and 'Live Share'.
- Status Bar:** Shows 'Ln 27, Col 34', 'Spaces: 4', 'UTF-8', 'CRLF', 'HTML', 'Go Live', 'Prettier', and system status (ENG IN 1/19/2022).
- Bottom Icons:** Standard Windows-style taskbar icons for File, Search, Copy, Paste, etc.

File Edit Selection View Go Run Terminal Help

signup.html - Buzzcars - Visual Studio Code

BUZZCARS

- BUZZCARS
  - > Buzzcars
  - > media
  - > src
  - > static
  - templates
    - > about.html
    - > index.html
    - > login.html
    - > signup.html
    - > vehicles.html
  - db.sqlite3
  - manage.py
  - > env
  - Synopsis.md

signup.html

```
28     <ul class="messages">
29         {% for message in messages %}
30             <p>{{ message }}</p>
31         {% endfor %}
32     {% endif %}
33     <form method='POST' action="">
34         {% csrf_token %}
35         <div class="inputbx">
36             <span>Name</span>
37             {{ form.username }}
38         </div>
39         <div class="inputbx">
40             <span>Email</span>
41             {{ form.email }}
42         </div>
43         <div class="inputbx">
44             <span>Password</span>
45             {{ form.password1 }}
46         </div>
47         <div class="inputbx">
48             <span>Confirm Password</span>
49             {{ form.password2 }}
50         </div>
51         <div class="inputbx">
52             <input type="submit" value="Sign Up">
53         </div>
54     <div class="txt"><h4>include digits in your password</h4></div>
55
56
57     </form>
58
59 </div>
60 </section>
61 </body>
62 </html>
```

Python 3.9.6 64-bit ('env': venv) 0 0 0 Rahul Live Share

File Edit Selection View Go Run Terminal Help

vehicles.html - Buzzcars - Visual Studio Code

BUZZCARS

- BUZZCARS
  - > Buzzcars
  - > media
  - > src
  - > static
  - templates
    - > about.html
    - > index.html
    - > login.html
    - > signup.html
    - > vehicles.html
  - db.sqlite3
  - manage.py
  - > env
  - Synopsis.md

vehicles.html

```
1  {% load static %}
2  <!DOCTYPE html>
3  <html lang="en">
4      <head>
5          <meta charset="UTF-8">
6          <meta http-equiv="X-UA-Compatible" content="IE=edge">
7          <meta name="viewport" content="width=device-width, initial-scale=1.0">
8              Follow link (ctrl + click)
9          <link rel="stylesheet" href="https://unpkg.com/swiper@7/swiper-bundle.min.css" />
10
11         <!-- font awesome cdn link -->
12         <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css" />
13
14         <!-- custom css file link -->
15         <link rel="stylesheet" href="{% static 'css/style.css' %}" />
16
17     </head>
18     <body>
19         <header class="header">
20             <div id="menu-btn" class="fas fa-bars"></div>
21             <a href="#" class="logo"> Buzzcars </a>
22
23             <nav class="navbar">
24                 <a href="{% url 'home' %}"> home </a>
25                 <a href="{% url 'vehicle' %}"> vehicles </a>
26                 <a href="{% url 'home' %}#services"> services </a>
27                 <a href="{% url 'about' %}"> About Us </a>
28                 <a href="{% url 'home' %}#contact"> Contact Us </a>
29             </nav>
30
31         </header>
32
33         <div class="grid-container">
34             {% for vehicle in vehicles %}
35                 <div class="grid-item">
36                     <div class="card">
37                         
38                         <div class="card__content">
```

Python 3.9.6 64-bit ('env': venv) 0 0 0 Rahul Live Share

File Edit Selection View Go Run Terminal Help

vehicles.html - Buzzcars - Visual Studio Code

BUZZCARS

- BUZZCARS
  - > Buzzcars
  - > media
  - > src
  - > static
  - templates
    - > about.html
    - > index.html
    - > login.html
    - > signup.html
    - > vehicles.html
  - db.sqlite3
  - manage.py
  - > env
  - Synopsis.md

vehicles.html

```
1  {% load static %}
2  <!DOCTYPE html>
3  <html lang="en">
4      <head>
5          <meta charset="UTF-8">
6          <meta http-equiv="X-UA-Compatible" content="IE=edge">
7          <meta name="viewport" content="width=device-width, initial-scale=1.0">
8              Follow link (ctrl + click)
9          <link rel="stylesheet" href="https://unpkg.com/swiper@7/swiper-bundle.min.css" />
10
11         <!-- font awesome cdn link -->
12         <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css" />
13
14         <!-- custom css file link -->
15         <link rel="stylesheet" href="{% static 'css/style.css' %}" />
16
17     </head>
18     <body>
19         <header class="header">
20             <div id="menu-btn" class="fas fa-bars"></div>
21             <a href="#" class="logo"> Buzzcars </a>
22
23             <nav class="navbar">
24                 <a href="{% url 'home' %}"> home </a>
25                 <a href="{% url 'vehicle' %}"> vehicles </a>
26                 <a href="{% url 'home' %}#services"> services </a>
27                 <a href="{% url 'about' %}"> About Us </a>
28                 <a href="{% url 'home' %}#contact"> Contact Us </a>
29             </nav>
30
31         </header>
32
33         <div class="grid-container">
34             {% for vehicle in vehicles %}
35                 <div class="grid-item">
36                     <div class="card">
37                         
38                         <div class="card__content">
```

Python 3.9.6 64-bit ('env': venv) 0 0 0 Rahul Live Share

File Edit Selection View Go Run Terminal Help

EXPLORER BUZZCARS

vehicles.html

```
31     </nav>
32   </header>
33   <div class="grid-container">
34     {% for vehicle in vehicles %}
35       <div class="grid-item">
36         <div class="card">
37           
38           <div class="card__content">
39             <h1 class="card__title">{{vehicle.name}}</h1>
40             <h3 class="card__title"><span class="fas fa-circle"></span> {{vehicle.type}} <span class="fas fa-circle"></span> {{vehicle.year}}</h3>
41             <p>{{vehicle.description}}</p>
42             <div class="card__price">Rs.{{vehicle.cost}}</div>
43             <div class="card__content"><a href="{{ url 'home' }}#contact" class="btw">Book a Quote</a></div>
44           </div>
45         </div>
46       </div>
47     {% endfor %}
48   </div>
49 <script src="https://unpkg.com/swiper@7/swiper-bundle.min.js"></script>
50 <script src="{{ static 'JS/script.js' }}></script>
51 </body>
52 </html>
```

OUTLINE

Python 3.9.6 64-bit ('env': venv) 0 0 0 Live Share

Ln 13, Col 1 Spaces: 4 UTF-8 CRLF HTML Go Live Prettier ENG 8:40 PM IN 1/19/2022