

## Table of Contents

1.	Introduction:.....	5
2.	Research and Planning:.....	5
3.	Choice of Framework and Technologies: .....	5
3.1.	Python:.....	5
3.2.	Django Framework: .....	5
3.3.	PostgreSQL: .....	5
3.4.	Bootstrap Template:.....	6
3.5.	Heroku:.....	6
4.	UX Design:.....	7
4.1.	Flow Diagram:.....	7
4.2.	Homepage:.....	8
4.3.	Homepage (Featured Cars):.....	9
4.4.	Homepage (Latest Cars):.....	10
4.5.	Homepage (Team):.....	11
4.6.	Login Page: .....	11
4.7.	Register Page:.....	12
4.8.	Car Page: .....	14
4.9.	Single Car Page: .....	15
4.10.	About Page:.....	16
4.11.	Services Page:.....	17
4.12.	Search model:.....	18
4.13.	Search result Page:.....	19
4.14.	Backend.....	19
5.	Web Services: .....	24
5.1.	Login using Facebook:.....	24
5.2.	Login using Google: .....	26
6.	Security Threats and Measures:.....	28
6.1.	CSRF (Cross-site request forgery): .....	28
7.	Web Application Features:.....	28

8.	Links:.....	31
9.	References: .....	32

### **Figure Table:**

Figure 4.1.1.: Carzone Web Application Flow Diagram.....	7
Figure 4.2.1: Homepage (Main).....	8
Figure 4.3.1.: Homepage (Featured Cars):.....	9
Figure 4.3.2.: Selected Car photo library. ....	9
Figure 4.4.1.: Homepage (Latest Cars) .....	10
Figure 4.5.1.: Homepage (Team) .....	11
Figure 4.6.1: Login for users. ....	12
Figure 4.7.1.: Register Page.....	12
Figure 4.7.2.: Incorrect password Error.....	13
Figure 4.7.3.: Username Error.....	13
Figure 4.7.4.: Email Error.....	13
Figure 4.8.1.: Car's page .....	14
Figure 4.9.1.: Single Car detail page. ....	15
Figure 4.9.2.: Single car description page.....	15
Figure 4.9.2.: Single car Feature page. ....	16
Figure 4.10.1.: About page.....	16
Figure 4.10.2.: About page.....	17
Figure 4.11.1.: Service Page .....	18
Figure 4.11.2.: Service Page .....	18
Figure 4.12.1.: search display.....	18
Figure 4.13.1.: search result page .....	19
Figure 4.14.1: PostgreSQL setting in the setting file.....	19
Figure 4.14.2: Admin Login Page.....	20

Figure 4.14.3: Django Admin Page.....	21
Figure 4.14.4: Add car details page.....	22
Figure 4.14.5: Add car details page.....	23
Figure 4.14.6: Add car details page.....	23
Figure 4.14.7: Contact tab.....	24
Figure 4.14.8: Team.....	24
Figure 5.1.1.: Facebook developer's page.....	25
Figure 5.1.2.: Facebook login setting in the admin page.....	26
Figure 5.1.3.: href link on the HTML page.....	26
Figure 5.2.1.: Google cloud API credential settings page.....	27
Figure 5.2.2.: Google href setting in the HTML page.....	27
Figure 6.1.1: csrf token.....	28
Figure 7.1.1: Search button in the navigation bar.....	28
Figure 7.1.2: Log in and Register Feature.....	29
Figure 7.1.3: Login Feature.....	29
Figure 7.1.4: Dashboard Feature.....	29
Figure 7.1.5: Facebook and Google Login Feature.....	30
Figure 7.1.6: Send Inquiry message dialogue box.....	31

## **1. Introduction:**

Carzone is an interactive e-commerce solution providing users with an opportunity to buy pre-used cars. In Ireland, there are a plethora of car dealerships for secondhand cars. Carzone is the only car dealership in Ireland that deals in only luxury cars. Carzone provides users with a wide range of luxury cars to select from. The users can search for their dream car on Carzone at a very reasonable price at their nearest location. The user can also register themselves with the web application. The web application also has the option for the users to login with google and Facebook. Carzone delivers cars all over Ireland and to Northern Ireland. The user has the option to inquire in Carzone if they cannot find their dream car on the web application. Other than selling cars Carzone also provides multiple services to users who purchase cars from them. Services include Free Support.

## **2. Research and Planning:**

The web application Carzone is created for users who love cars, especially who love luxury cars. There are multiple car-selling websites in Ireland, but the luxury ones are few. To start with its development Django Python is used for the framework. To create a web application that is both functional on a desktop and a mobile device bootstrap is used. PostgreSQL is used for the backend of this web application.

## **3. Choice of Framework and Technologies:**

### **3.1. *Python:***

Python version 3.9.6 has been implemented in this web application. Python is used to construct web applications on the server. This web application is purely developed using python Django. Python is used throughout the project. The settings, files, and data models in the web application are developed using python. the web application is developed in a virtual environment.

### **3.2. *Django Framework:***

Django is a python based free and open-source web framework which helps developers to easily create complex and database-driven web applications. Django framework promotes the use of code reusability, rapid development, clean, and pragmatic design for the developers. Django supports five database backends which are PostgreSQL, MySQL, Oracle, Microsoft SQL, and MariaDB. All the installed applications, external libraries, backend settings, debug settings, etc., of the web application are developed in the ‘settings.py’ file main project file.

### **3.3. *PostgreSQL:***

Postgres is a relational database management system (RDBMS) that focuses on flexibility and SQL conformance. It is a free database management system. Transactions with Atomicity, Consistency, Isolation, and Durability (ACID) features, constantly updates views, materialized views, triggers, foreign keys, and stored procedures all are available in PostgreSQL. It can perform a huge spectrum of workloads, from single workstations to data

warehouses or Web services with a significant number of concurrent users. It is the default database for macOS servers, as well as Windows, Linux, FreeBSD, and OpenBSD (PostgreSQL - Wikipedia, 2022).

### ***3.4. Bootstrap Template:***

Bootstrap is a dynamic front-end framework for developing modern web pages and web applications. It's free and open to use, although it comes with lots of HTML and CSS templates for UI elements including buttons and forms. JavaScript extensions are also supported by Bootstrap. Because Bootstrap templates are pre-styled, users don't have to worry about making a bunch of designs. In Bootstrap, templates are appealing, modern, and, most importantly, simple to interact with. In a bootstrap template, the users need to make changes according to what they want in the web application and discard the changes which come with the templates.

### ***3.5. Heroku:***

Heroku is a cloud Platform as a Service (PaaS) that operates on containers (PaaS). Heroku is a platform that allows developers to deploy, manage, and scale modern programs. Our platform is attractive, adaptable, and simple to use, making it the quickest way for developers to get their products to market.

## 4. UX Design:

### 4.1. Flow Diagram:

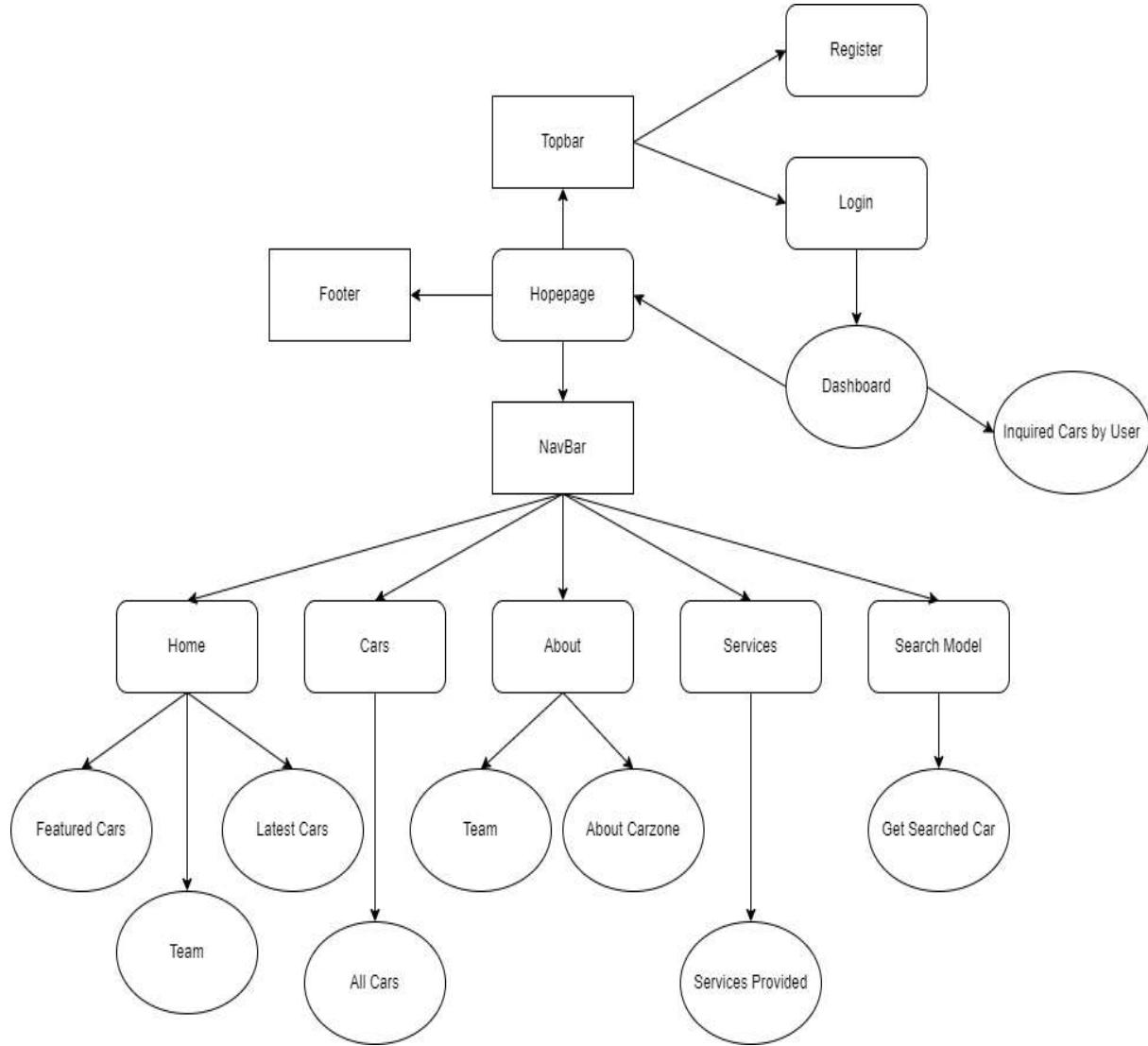


Figure 4.1.1.: Carzone Web Application Flow Diagram

## 4.2. Homepage:

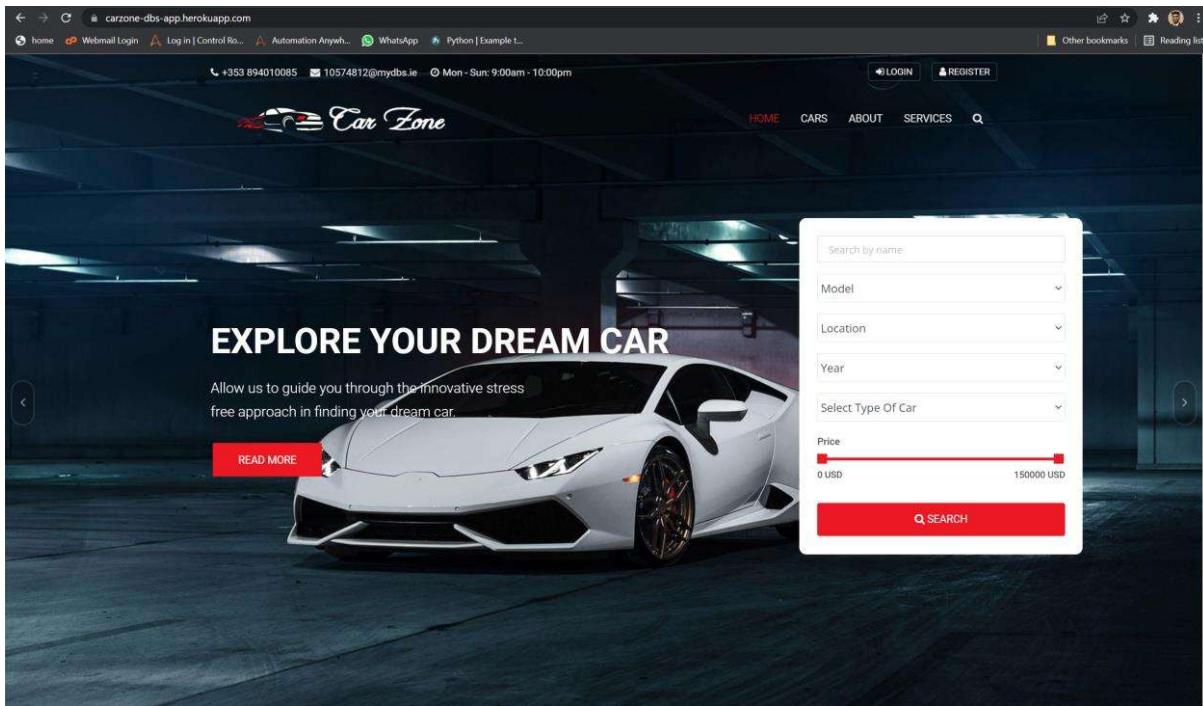


Figure 4.2.1: Homepage (Main)

The homepage consists of four sections. The top bar of the homepage shows the user multiple details. Carzone's contact number, email id, timings of the showroom, login, and register model. The navbar, which is below the top bar gives the user choices to navigate through the home page, cars page, about page, service page, and search page. The user can also search through all the cars given on the website through the search bar provided on the homepage. The user can filter the search panel through the car model, location, year of the car, type of the car, and price.

#### 4.3. Homepage (Featured Cars):

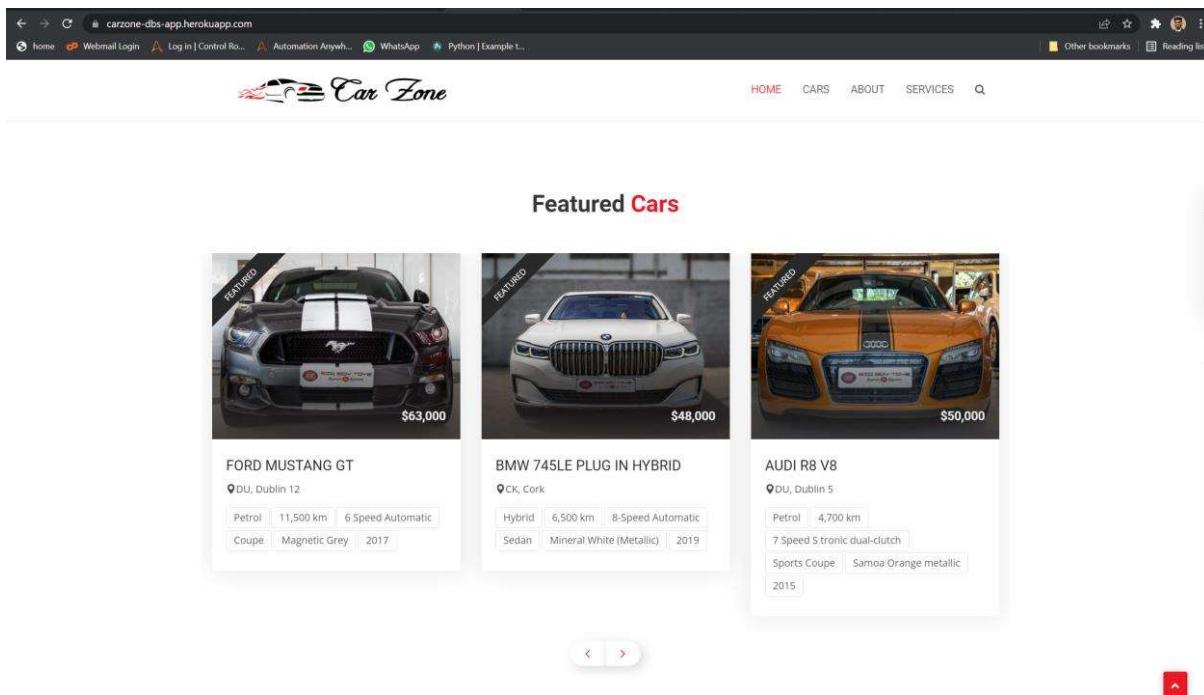


Figure 4.3.1.: Homepage (Featured Cars):

The Homepage has a section for all the featured cars. The featured cars section shows all the top cars available with Carzone. The user can check the highlighted features of the car on the card of that car or can also get all the details of the car by clicking on the car name.



Figure 4.3.2.: Selected Car photo library.

The user can also browse through all the car photos from the selected car by clicking on the thumbnail of the car.

#### 4.4. Homepage (Latest Cars):

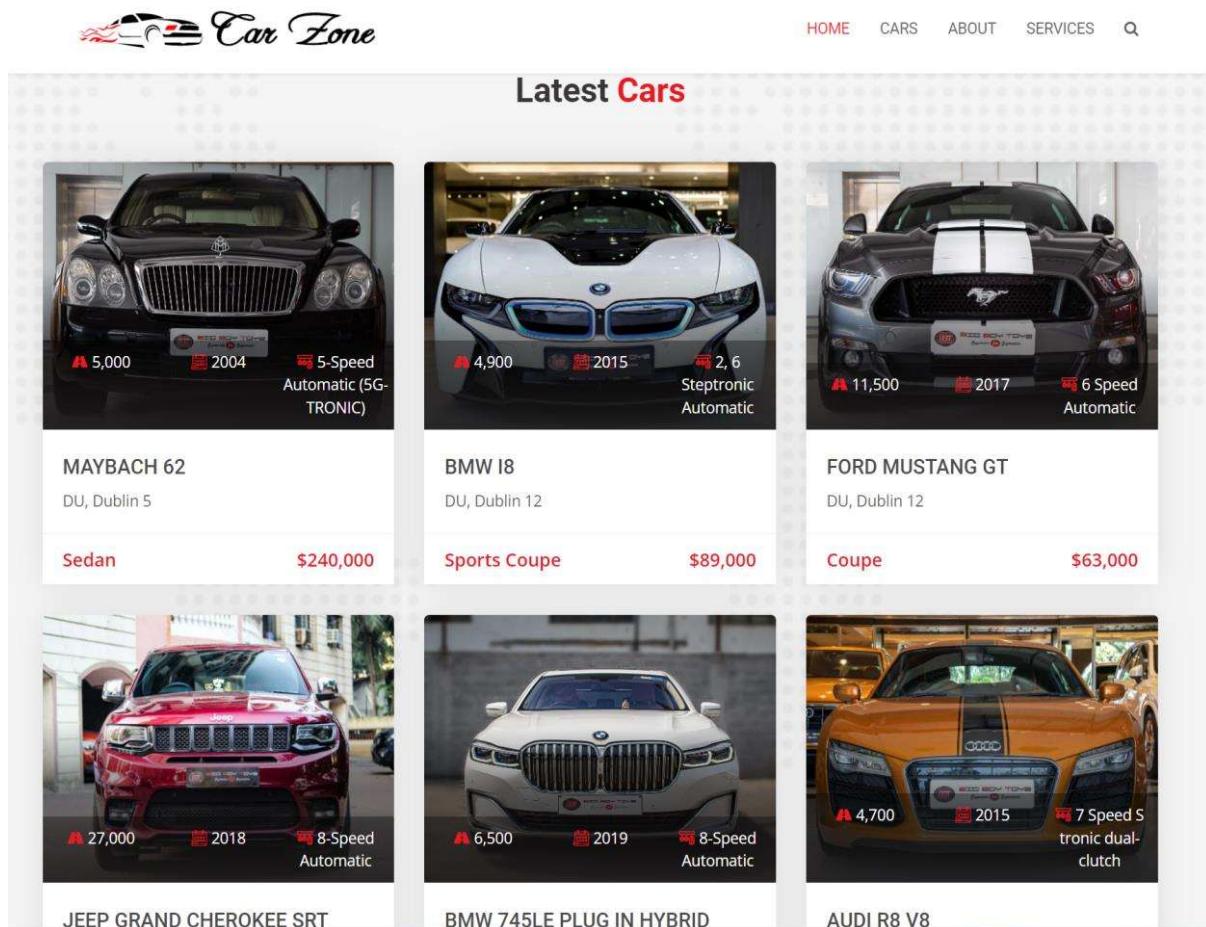


Figure 4.4.1.: Homepage (Latest Cars)

In this section, the user can browse through all the cars which have been added to the website lately. It has the same functionality as the featured cars.

#### 4.5. Homepage (Team):

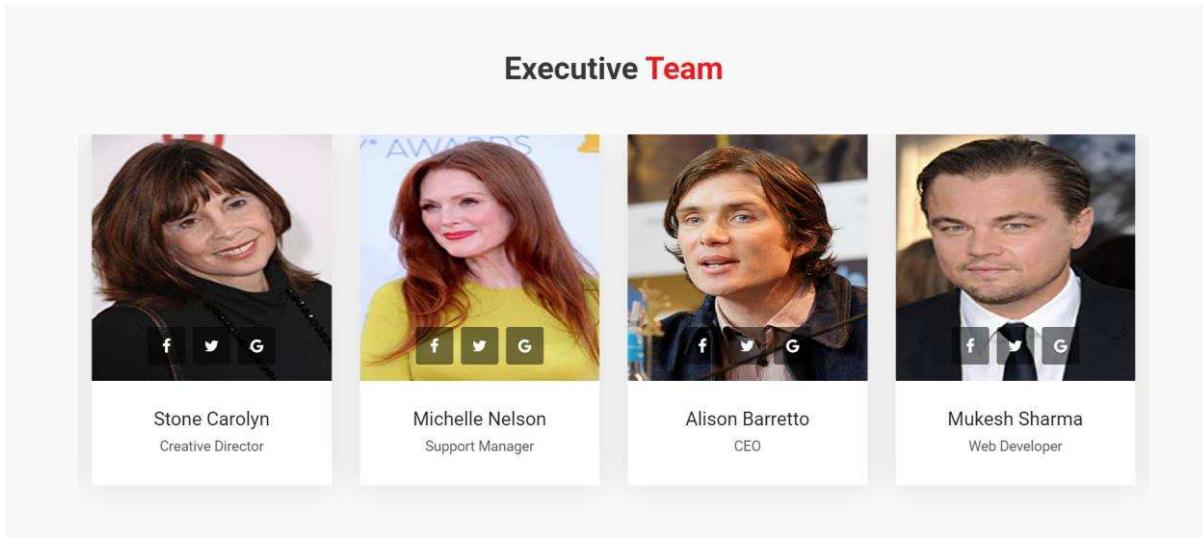


Figure 4.5.1.: Homepage (Team)

The end of the homepage consists of the Carzone team members and their designation.

#### 4.6. Login Page:

A screenshot of the CarZone login page. At the top is the CarZone logo, which features a stylized car icon followed by the word "Car Zone" in a script font. Below the logo is the text "Sign into your account". There are two input fields: one for "Username" with a mail icon and one for "Password" with a lock icon. A red "LOGIN" button is positioned below the password field. Below the login form is a horizontal line with the text "Or Login With" in the center. To the left of the line is a blue "Facebook" button with a white "f" icon, and to the right is a red "Google" button with a white "G" icon. At the bottom of the page is a link "Don't have an account? Register here".

Figure 4.6.1: Login for users.

In the above figure, the user can log in if he or she is an existing user. The user can also log in using Facebook or Google accounts. If the user is new the user can click on the register button to create a new account in Carzone.

#### 4.7. Register Page:

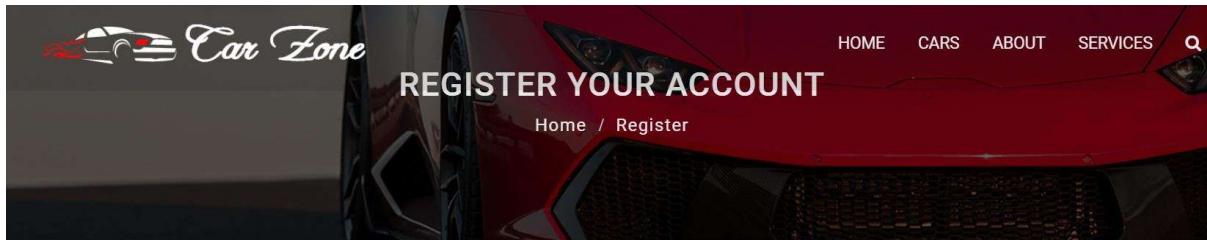
A screenshot of the "Create an account" registration form. It includes fields for First Name, Last Name, Username, Email Address, Password, and Confirm Password, each with a corresponding icon. A red "REGISTER" button is at the bottom.

Figure 4.7.1.: Register Page.

On this page, the new user can register themselves by creating an account with Carzone. The user can register themselves using Facebook and Google accounts as well. All the fields are required by the user to be filled. There are certain validations used at this point. The user will be prompted a message if any of the validation fails to match, such as if the username exists in the database, then the user needs to change its username. If an email id also exists in the database, then the user will be asked to change the email id as well. Password and confirm password need to be the same as well when registering themselves.

## Create an account

ERROR: PASSWORD DO NOT MATCH X

First Name

Figure 4.7.2.: Incorrect password Error.



**Car Zone**

Create an account

ERROR: USERNAME ALREADY EXISTS! X

Figure 4.7.3.: Username Error.



**Car Zone**

Create an account

ERROR: EMAIL ALREADY EXISTS! X

Figure 4.7.4.: Email Error.

#### 4.8. Car Page:

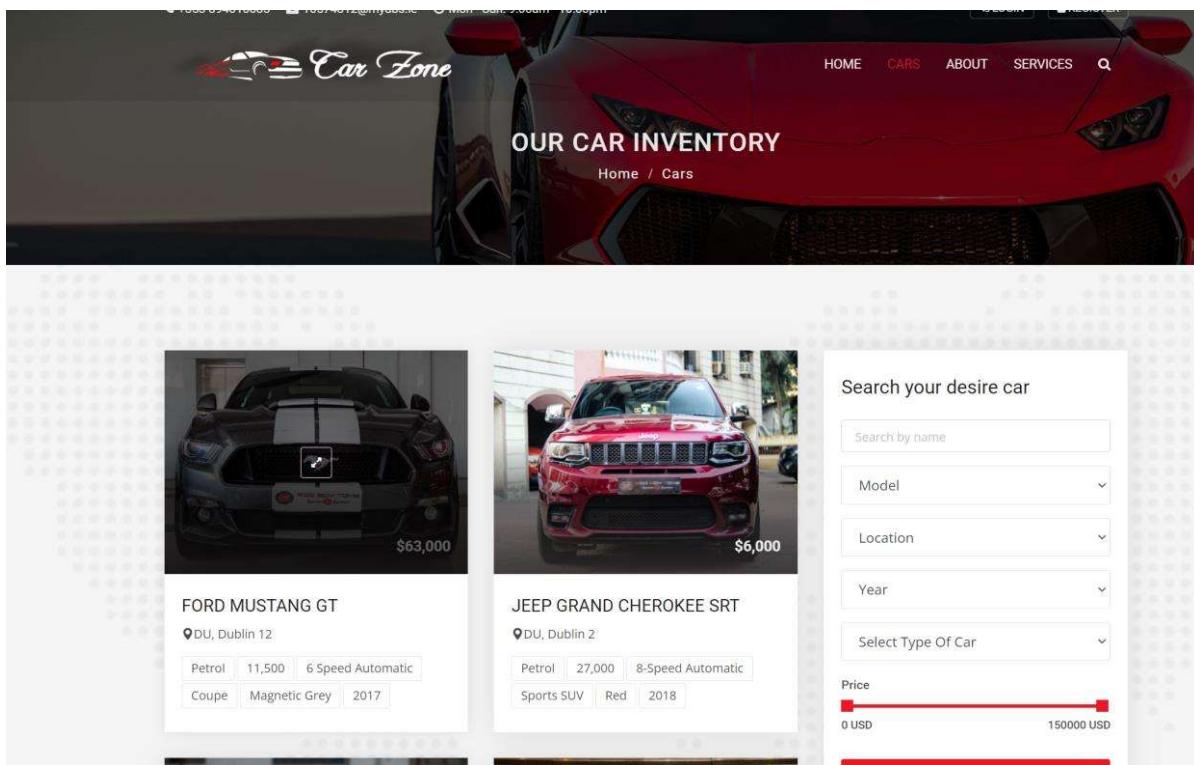


Figure 4.8.1.: Car's page

On this page, the user can browse through all the cars the page can offer. The user can also search the cars based on the model, location, year, and price as well. On selecting an of the car model, the user will be redirected to the car details page where all the details of the car are provided.

#### 4.9. Single Car Page:

**FORD MUSTANG GT** 63,000

📍 DU, Dublin 12

Refine Your Search

Color:	Magnetic Grey
Model:	MUSTANG GT
Year:	2017
Condition:	Used
Body Style:	Coupe
Engine:	5.0 L, V8, Longitudinal
Transmission:	6 Speed Automatic
Interior:	Black Leather
Miles:	11,500 km
Doors:	4
Passengers:	4
Vin Number:	234DFTGYHUJ
Mileage:	10
Fuel Type:	Petrol
Owners:	2

**Vehicle overview** **Features**

Figure 4.9.1.: Single Car detail page.

On this page, the user is provided with all the details of the car. The user can browse through all the details such as the color, model, year, condition, body style, engine, transmission, interior, Miles, doors, passengers, VIN, mileage, fuel type, owners, vehicle overview, and all the features of the car.

**Vehicle overview** **Features**

**Description**

The **sixth generation Ford Mustang (S550)** is the current iteration of the Mustang pony car manufactured by Ford. In departure from prior Mustang models, the sixth generation Mustang includes fully independent rear suspension on all models, as well as an optional 2.3L EcoBoost turbocharged and direct injected four-cylinder engine. The new Mustang was introduced as a 2015 model year vehicle, marking the fiftieth anniversary of the Ford Mustang, which was revealed as a 1965 model year vehicle on April 17, 1964. The sixth generation is also the first Ford Mustang to be marketed and sold globally, and represented the first time that factory right hand drive Mustangs were produced in addition to the left hand drive models. This is part of the "One Ford" business plan, which also applies to the Fiesta, Focus, Fusion/Mondeo, Escape/Kuga, Edge, Ford Transit Connect, and Ford Transit, as well as other models.

**Fuel Type:** Petrol  
**Owners:** 2

**Get a Question?**

- 📍 Dublin Business School
- ✉ 10574812@mydbs.ie
- 📞 +353 0894010086

Figure 4.9.2.: Single car description page.

The screenshot shows a car listing page. At the top, there are tabs for "Vehicle overview" and "Features", with "Features" being the active tab. Below the tabs, the word "Features" is displayed in bold. To the right, there is a table with two rows. The first row contains "Fuel Type: Petrol" and "Owners: 2". The second row is partially visible. On the left side, there is a list of vehicle features: Airbags, Air Conditioning, Seat Heating, Alarm System, ParkAssist, Power Steering, Reversing Camera, Direct Fuel Injection, and Auto Start/Stop. On the right side, there is a section titled "Get a Question?" with icons for location, email, and phone, followed by the text "Dublin Business School", "10574812@mydbs.ie", and "+353 0894010086". Below this are social media sharing buttons for Facebook, Twitter, Google+, RSS, and LinkedIn.

Figure 4.9.2.: Single car Feature page.

#### *4.10. About Page:*

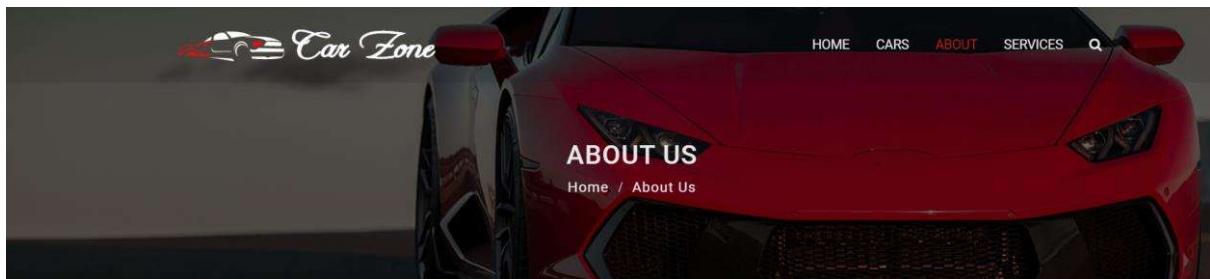


Figure 4.10.1.: About page.

On this page, the user is provided some description about Carzone and the team members of Carzone.

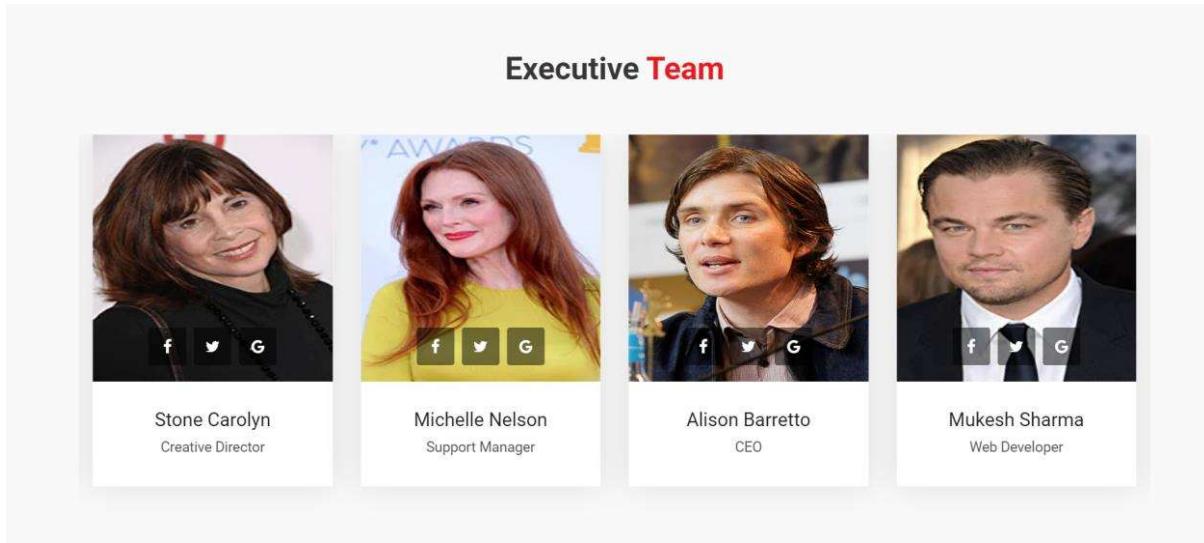
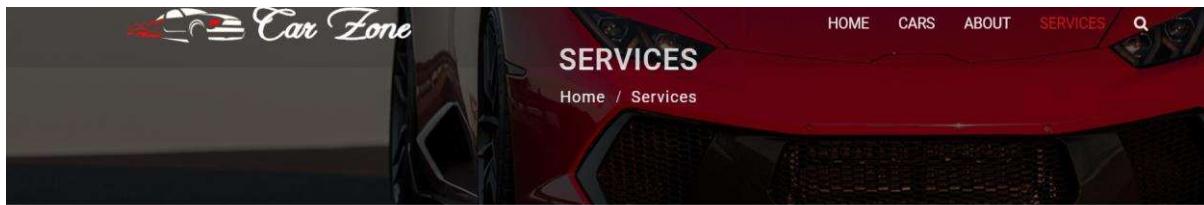


Figure 4.10.2.: About page.

#### 4.11. Services Page:



The screenshot shows a section titled "Our Services" with seven numbered items (1-7) each with an icon and a brief description.

Icon	Service	Description
	Free Support	Cars comes with complete previous service history checked with authorised dealership
	Super Fast	Guranteed fast cars according to the service history of the car
	Repairing	We provide free repair for the first 1 year of the purchased car.
	AIR conditioning	Free Air condition checks provided after every six months for 2 years.
	Oil Change	Free Oil checks and changes provided after every six months for 2 years.
	Engine Repair	All cars that we sell comes with a backed up six months warranty on engine and

Figure 4.11.1.: Service Page

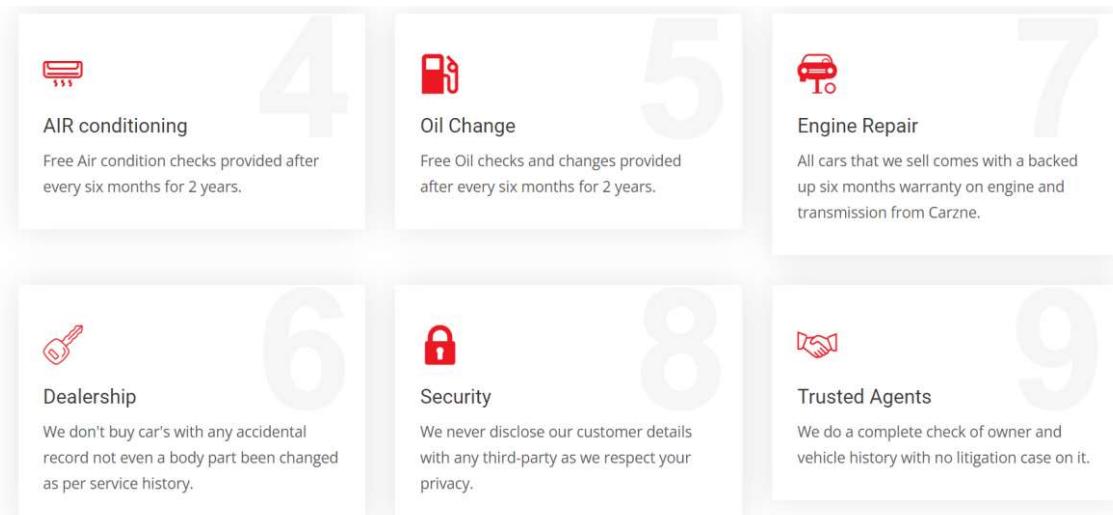


Figure 4.11.2.: Service Page

On this page, the user has shown all the services provided by Carzone.

#### 4.12. Search model:

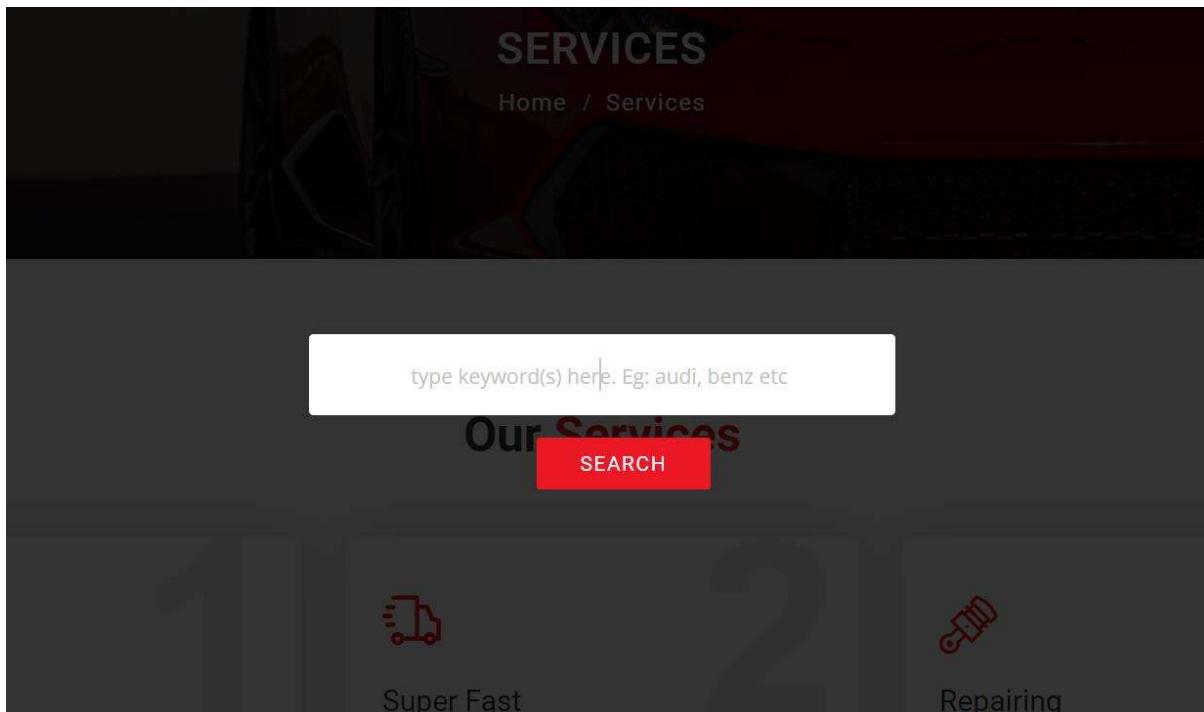


Figure 4.12.1.: search display

In the above figure, the user is prompted with a search box where the user can search for any cars on the website. On clicking the search button the user is redirected to the search result page.

#### 4.13. Search result Page:

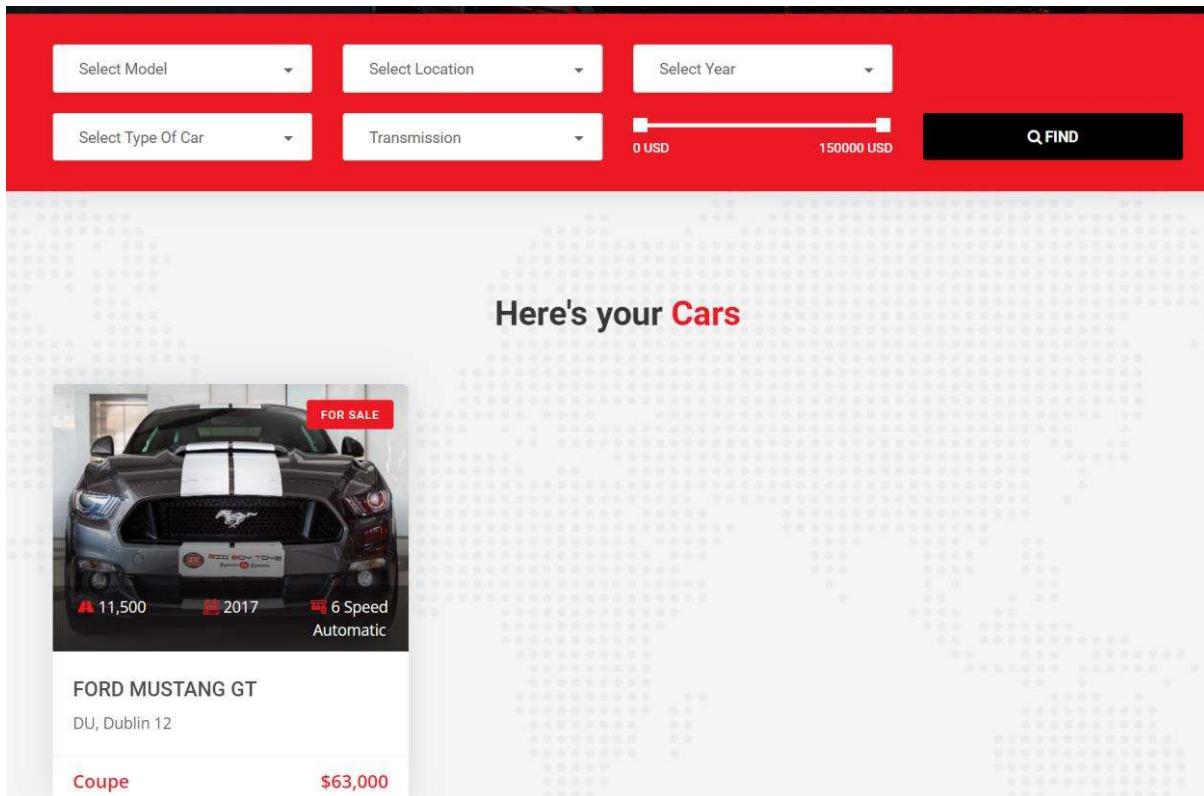


Figure 4.13.1.: search result page

Here the user is given the search result that he or she searched for.

#### 4.14. Backend:

As this web application uses the PostgreSQL database as its backend database, the `psycopg2` library is installed as it is an adaptor for PostgreSQL. The setting for the database has been configured in the `settings.py` file in the project's main folder.

```
90
91 # Database
92 # https://docs.djangoproject.com/en/3.0/ref/settings/#databases
93
94 DATABASES = {
95     'default': {
96         'ENGINE': 'django.db.backends.postgresql',
97         'NAME': 'carzone_db',
98         'USER': 'postgres',
99         'PASSWORD': 'Qwerty@123',
100        'HOST': 'localhost',
101    }
102}
103
104
105 DATABASES = {'default': dj_database_url.config(default='postgres://postgres:Qwerty@123@localhost/carzone_db')}
106
107
```

Figure 4.14.1: PostgreSQL setting in the setting file.

Data such as car data, team data, people inquiry is monitored by the Django admin page.

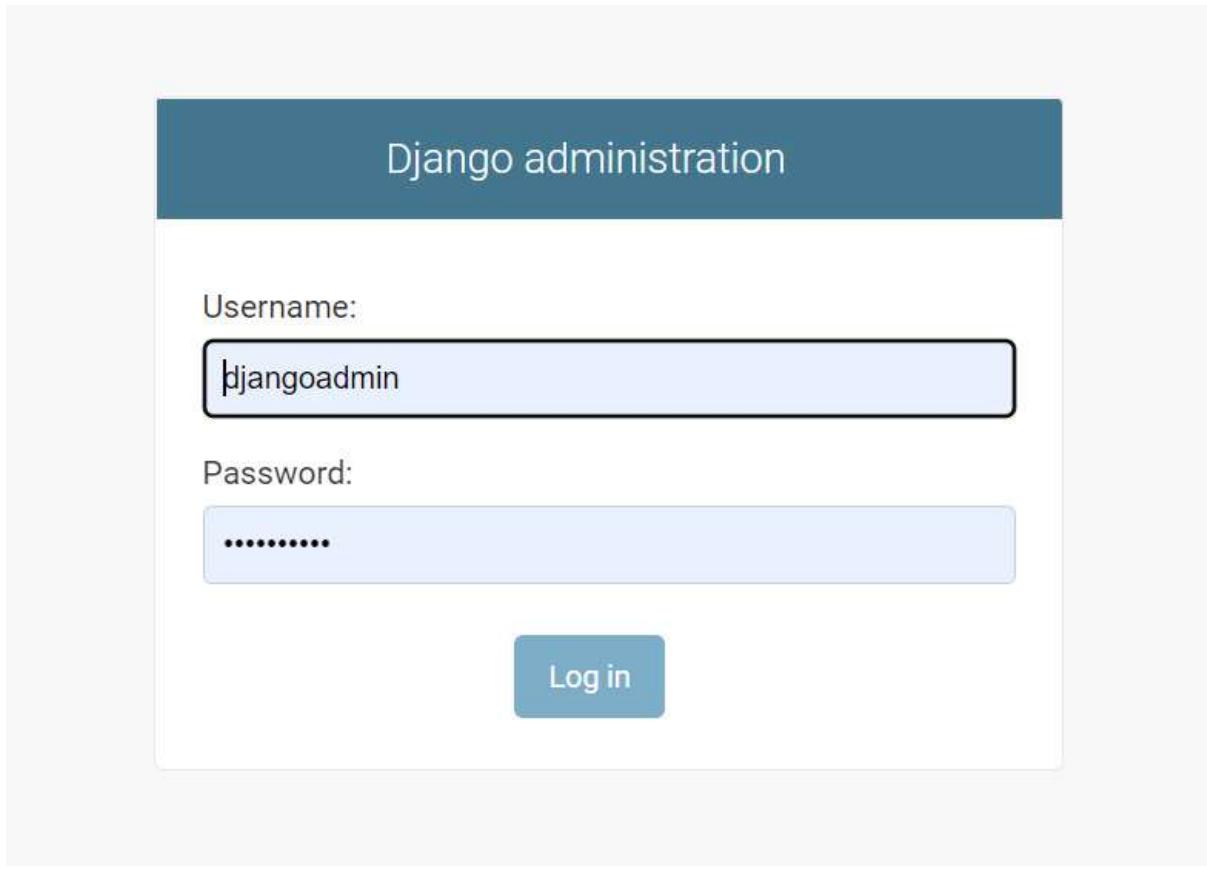


Figure 4.14.2: Admin Login Page.

To access the admin page we need to use the given link - "<https://carzone-dbs-app.herokuapp.com/admin/login/?next=/admin/>". The username and password are the same which is used in the local PostgreSQL login.

Django administration		
Site administration		
ACCOUNTS		
Email addresses	<a href="#"> Add</a>	<a href="#"> Change</a>
AUTHENTICATION AND AUTHORIZATION		
Groups	<a href="#"> Add</a>	<a href="#"> Change</a>
Users	<a href="#"> Add</a>	<a href="#"> Change</a>
CARS		
Cars	<a href="#"> Add</a>	<a href="#"> Change</a>
CONTACTS		
Contacts	<a href="#"> Add</a>	<a href="#"> Change</a>
PAGES		
Teams	<a href="#"> Add</a>	<a href="#"> Change</a>
SITES		
Sites	<a href="#"> Add</a>	<a href="#"> Change</a>
SOCIAL ACCOUNTS		
Social accounts	<a href="#"> Add</a>	<a href="#"> Change</a>
Social application tokens	<a href="#"> Add</a>	<a href="#"> Change</a>
Social applications	<a href="#"> Add</a>	<a href="#"> Change</a>

Figure 4.14.3: Django Admin Page.

On this page, the admin can view all the data and add the data needed for the web application.

Add car model:

Add car

Figure 4.14.4: Add car details page.

<b>Car photo:</b>	<input type="button" value="Choose File"/> No file chosen
Car photo 1:	<input type="button" value="Choose File"/> No file chosen
Car photo 2:	<input type="button" value="Choose File"/> No file chosen
Car photo 3:	<input type="button" value="Choose File"/> No file chosen
Car photo 4:	<input type="button" value="Choose File"/> No file chosen
<b>Features:</b>	<input type="checkbox"/> Cruise Control <input type="checkbox"/> Audio Interface <input type="checkbox"/> Airbags <input type="checkbox"/> Air Conditioning <input type="checkbox"/> Seat Heating <input type="checkbox"/> Alarm System <input type="checkbox"/> ParkAssist <input type="checkbox"/> Power Steering <input type="checkbox"/> Reversing Camera <input type="checkbox"/> Direct Fuel Injection <input type="checkbox"/> Auto Start/Stop <input type="checkbox"/> Wind Deflector <input type="checkbox"/> Bluetooth Handset
<b>Body style:</b>	<input type="text"/>
<b>Engine:</b>	<input type="text"/>
<b>Transmission:</b>	<input type="text"/>
<b>Interior:</b>	<input type="text"/>
<b>Miles:</b>	<input type="text"/>

Figure 4.14.5: Add car details page.

---

<b>Car photo:</b>	<input type="button" value="Choose File"/> No file chosen
Car photo 1:	<input type="button" value="Choose File"/> No file chosen
Car photo 2:	<input type="button" value="Choose File"/> No file chosen
Car photo 3:	<input type="button" value="Choose File"/> No file chosen
Car photo 4:	<input type="button" value="Choose File"/> No file chosen

---

<b>Features:</b>	<input type="checkbox"/> Cruise Control <input type="checkbox"/> Audio Interface <input type="checkbox"/> Airbags <input type="checkbox"/> Air Conditioning <input type="checkbox"/> Seat Heating <input type="checkbox"/> Alarm System <input type="checkbox"/> ParkAssist <input type="checkbox"/> Power Steering <input type="checkbox"/> Reversing Camera <input type="checkbox"/> Direct Fuel Injection <input type="checkbox"/> Auto Start/Stop <input type="checkbox"/> Wind Deflector <input type="checkbox"/> Bluetooth Handset
------------------	--

---

<b>Body style:</b>	<input type="text"/>
<b>Engine:</b>	<input type="text"/>
<b>Transmission:</b>	<input type="text"/>
<b>Interior:</b>	<input type="text"/>
<b>Miles:</b>	<input type="text"/>

---

Figure 4.14.6: Add car details page.

To add cars to the web application the admin needs to fill out the above form on the admin page.

Django administration

Home · Contacts · Contacts

Select contact to change

Search

Action:  Go 0 of 4 selected

<input type="checkbox"/>	ID	FIRST NAME	LAST NAME	EMAIL	CAR TITLE	CITY	CREATE DATE
<input type="checkbox"/>	11	Alison	barreto	alisonbarreto2123@gmail.com	FORD MUSTANG GT	MUMBAI	March 17, 2022, 11:52 p.m.
<input type="checkbox"/>	10	Alison	barreto	alisonbarreto2123@gmail.com	FORD MUSTANG GT	MUMBAI	March 17, 2022, 11:52 p.m.
<input type="checkbox"/>	9	Alison	barreto	alisonbarreto2123@gmail.com	FORD MUSTANG GT	MUMBAI	March 15, 2022, 2:49 p.m.
<input type="checkbox"/>	8	Alison	barreto	alisonbarreto2123@gmail.com	JEEP GRAND CHEROKEE SRT	MUMBAI	March 13, 2022, 3:17 p.m.

4 contacts

Figure 4.14.7: Contact tab.

To check all the inquiries from the user the admin needs to check the contacts tab.

Home · Pages · Teams

Select team to change

ADD TEAM +

Action:  Go 0 of 5 selected

<input type="checkbox"/>	TEAM
<input type="checkbox"/>	Mukesh
<input type="checkbox"/>	Michelle
<input type="checkbox"/>	Aman
<input type="checkbox"/>	Stone
<input type="checkbox"/>	Alison

5 teams

Figure 4.14.8: Team.

To add new team members or remove any team member the admin can use the team's tab.

## 5. Web Services:

### 5.1. Login using Facebook:

The user can log in or register themselves using their Facebook account. To start off with this Django-allauth library is installed on the machine.

Django-allauth: Multiple authentication methods (e.g., login by username or e-mail), as well as account verification strategies, as well as account verification strategies, are supported (ranging from none to e-mail verification) by this library. Multiple social accounts can be linked to a Django user account. The required consumer keys and secrets for connecting with Facebook, Twitter, and other social media platforms must be configured in the database using the SocialApp model by the Django admin. The Django sites framework is used for consumer keys and tokens. This is especially useful for bigger multi-domain projects, but it also allows you to switch between a development (localhost) and production setup without affecting your settings or database.

To add Facebook keys, the developer needs to create a new app on the developer's page Facebook. On creating the app, the developer needs to configure the app name and app secret id on the admin page. Also, the URL needs to be added to the Facebook app as well from where the API will be called from.

The screenshot shows the Facebook developer console interface. At the top, it displays the app ID '666155004816124' and the status 'In development'. The left sidebar has a 'Settings' section selected, with 'Basic' sub-selected. Other sections include 'Roles', 'Alerts', 'App Review', 'Products', 'Facebook Login', and 'Activity Log'. The main right panel shows the app's configuration details:

- App ID:** 666155004816124
- App secret:** (redacted)
- Display name:** Django Carsale
- Namespace:** (empty)
- App domains:** (empty)
- Contact email:** alisonbarreto2123@gmail.com
- Privacy Policy URL:** Privacy policy for Login dialog and app details
- Terms of Service URL:** Terms of Service for Login dialog and App Details
- App icon (1024 x 1024):** A placeholder icon for the app.
- Category:** Choose a category
- App purpose:** This app's primary purpose is to access and use data from Facebook's Platform on behalf of:
  - Yourself or your own business
  - Clients
 

Select this option if the primary purpose of this app is to manage data or assets on behalf of an individual client or multiple clients.

If you are developing an app that accesses and uses data from Facebook's Platform on behalf of clients, you are subject to

Figure 5.1.1.: Facebook developer's page.

The screenshot shows the Django administration interface under 'Social Accounts > Social applications > Facebook login'. The title bar says 'Django administration'.

The page is titled 'Change social application' and shows the following fields:

- Provider:** Facebook
- Name:** Facebook login
- Client id:** 666155004816124  
App ID, or consumer key
- Secret key:** b857597034b62a9813ddd9a969223762  
API secret, client secret, or consumer secret
- Key:** (empty)
- Sites:** A list of available sites including 'example.com' and 'carzone-dbs-app.herokuapp.com'.

Below the site lists are buttons: 'Choose all' and 'Remove all'. A note at the bottom says: 'Hold down "Control", or "Command" on a Mac, to select more than one.'

Figure 5.1.2.: Facebook login setting in the admin page.

The redirected path needs to be changed in the HTML file as well so that when the user clicks on the Facebook button on the login page he should be redirected to the login page of Facebook.

```
<ul class="social-list">
  <li><a href="{% provider_login_url 'facebook' method='js_sdk' %}" class="facebook-bg"><i class="fa fa-facebook facebook-i"></i><span>Facebook</span></a></li>
  <li><a href="{% provider_login_url 'google' %}" class="google-bg"><i class="fa fa-google google-i"></i><span>Google</span></a></li>
```

Figure 5.1.3.: href link on the HTML page.

The method used to authenticate here is js\_sdk. To accomplish this, we need to add the Heroku URL to the developer's page and toggle the login using the Javascript SDK button.

## 5.2. *Login using Google:*

The initial steps are the same as the ones used for Facebook login configuration. To use Google login features, an account needs to be created on the google cloud portal. For google authentication OAuth 2.0 method is used. the Heroku URL and the local URL need to be added on the google cloud API service page and the client ID and secret key need to be configured on the admin page in social accounts.

Google Cloud Platform Carzone Login Search Products, resources

**API** APIs & Services Client ID for Web application DOWNLOAD JSON RESET

- Enabled APIs & services
- Library
- Credentials**
- OAuth consent screen
- Domain verification
- Page usage agreements

Name \* Carzone  
The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

The domains of the URIs you add below will be automatically added to your [OAuth consent screen](#) as [authorized domains](#).

### Authorized JavaScript origins

For use with requests from a browser

URIs 1 \* http://localhost:8000  
URIs 2 \* http://127.0.0.1:8000  
URIs 3 \* https://carzone-dbs-app.herokuapp.com

+ ADD URI

### Authorized redirect URIs

For use with requests from a web server

URIs 1 \* http://localhost:8000/socialaccounts/google/login/callback/  
URIs 2 \* http://127.0.0.1:8000/socialaccounts/google/login/callback/  
URIs 3 \* https://carzone-dbs-app.herokuapp.com/socialaccounts/google/login/callback

+ ADD URI

Figure 5.2.1.: Google cloud API credential settings page.

```
<ul class="social-list">
<li><a href="{% provider_login_url 'facebook' method='js_sdk %}" class="facebook-bg"><i class="fa fa-facebook facebook-i"></i><span>Facebook</span></a></li>
<li><a href="{% provider_login_url 'google' %}" class="google-bg"><i class="fa fa-google google-i"></i><span>Google</span></a></li>
```

Figure 5.2.2.: Google href setting in the HTML page.

## 6. Security Threats and Measures:

### 6.1. CSRF (Cross-site request forgery):

What is CSRF?

Cross-site request forgery (commonly known as CSRF) is a web security flaw that allows an attacker to trick users into doing things they don't want to do. It allows an attacker to partially bypass the same-origin policy, which is intended to keep websites from interacting with one another.

How to avoid a CSRF attack?

A CSRF token is a one-of-a-kind, secret, and unpredictable value generated by the server-side application and sent to the client in such a way that it is included in the client's next HTTP request. When the later request is made, the server-side program checks that it has the expected token and rejects it if it is absent or invalid. By making it impossible for an attacker to generate a fully valid HTTP request suited for feeding to a victim user, CSRF tokens can help prevent CSRF attacks. Because the attacker cannot identify or predict the value of a user's CSRF token, they are unable to generate a request that has all of the parameters required for the application to honor the token.

```
<div class="login">...</div>
<h3>Sign into your account</h3>
<form action="/accounts/login" method="POST">
  <input type="hidden" name="csrfmiddlewaretoken" value="t6eSYgtsPiG9ocgv3WsWeIcwyOLEJUjtm2a4jBQyyxtv3Jk6xFrK0iNffz4HX0mr">
  <div class="form-group form-box">...</div>
  <div class="form-group form-box">...</div>
```

Figure 6.1.1: csrf token.

In the above figure, the value is the csrf token. We use the csrf token when we use a POST method in any web application.

## 7. Web Application Features:

Search Feature: the web application has a search button on all the pages in the navigation bar. On clicking this button, the user can browse through all the cars in the car inventory.

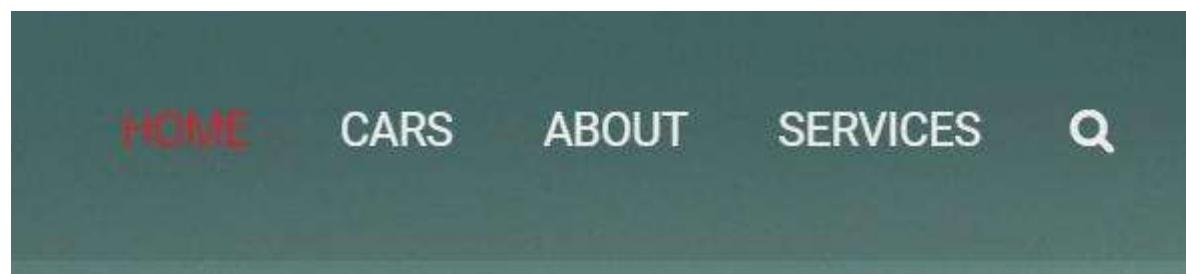


Figure 7.1.1: Search button in the navigation bar.

Login and register Feature: the user can log in using their existing username and password or by registering themselves using Facebook or googling the login feature. The user can also register themselves using the old fashion way of form filling.

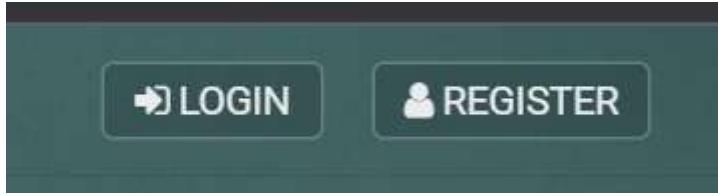


Figure 7.1.2: Log in and Register Feature.

**Logout Feature:** The web application contains a logout button once the user logs in. The user is redirected to the main homepage where the user will be asked to log in again.

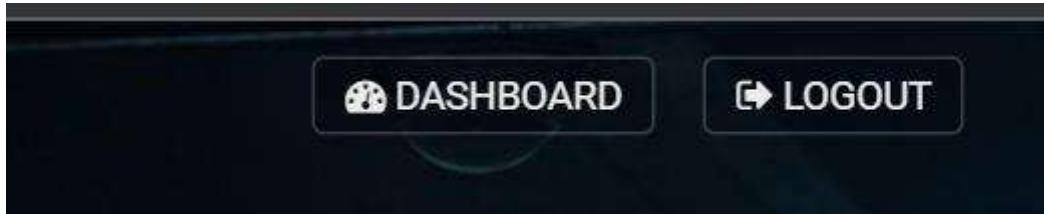
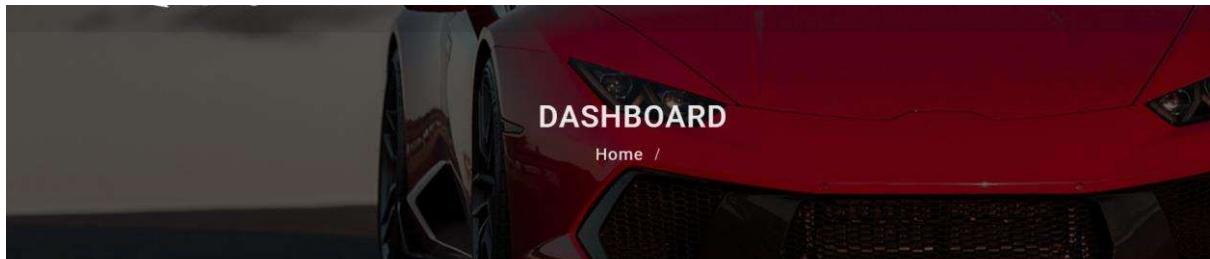


Figure 7.1.3: Login Feature.

**Dashboard Feature:**

The user who has created an account has access to a new page called dashboard. The user can view all the cars he/she has inquired about.



## Welcome

Here are the cars that you have inquired about

You have no inquiries

Figure 7.1.4: Dashboard Feature.

**Facebook and Google Login Feature:** The user can log in or create an account in the web application using their Facebook or Google accounts.



Sign into your account

Username

Password

**LOGIN**

Or Login With

Facebook

Google

Don't have an account? [Register here](#)

Figure 7.1.5: Facebook and Google Login Feature.

Inquiry Feature: The user can send a message using the send message button, which is available on every single car page, to send a message to the admin regarding anything about the car they want to buy. The car model will be dynamically shown in the model box.



Hello, My Name is

I'm interested in this

JEEP GRAND CHEROKEE SRT

I live in

You can reach me by email at

Add comments

Send

Figure 7.1.6: Send Inquiry message dialogue box.

Mobile Support Feature: the web application also functions well on a mobile screen as well. As the template used is developed using bootstrap it does not need many changes in the code.

## 8. Links:

GitHub Link: [https://github.com/alisonbarreto/carselling\\_dbsproject](https://github.com/alisonbarreto/carselling_dbsproject)

Heroku Link: <https://carzone-dbs.herokuapp.com>

## 9. References:

- Engineering Education (EngEd) Program | Section. 2022. *Creating a Django App Using PostgreSQL Database*. [online] Available at: <<https://www.section.io/engineering-education/django-app-using-postgresql-database/>> [Accessed 14 March 2022].
- En.wikipedia.org. 2022. *PostgreSQL - Wikipedia*. [online] Available at: <<https://en.wikipedia.org/wiki/PostgreSQL>> [Accessed 16 March 2022].
- Startbootstrap.com. 2022. *Bootstrap Themes vs. Templates - What's the Difference? - Start Bootstrap*. [online] Available at: <<https://startbootstrap.com/blog/bootstrap-themes-vs-templates>> [Accessed 15 March 2022].
- En.wikipedia.org. 2022. *Visual Studio Code - Wikipedia*. [online] Available at: <[https://en.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://en.wikipedia.org/wiki/Visual_Studio_Code)> [Accessed 16 March 2022].
- PyPI. 2022. *dj-database-url*. [online] Available at: <<https://pypi.org/project/dj-database-url/>> [Accessed 17 March 2022].
- Django, L., 2022. *Django-allauth Tutorial*. [online] Learndjango.com. Available at: <<https://learndjango.com/tutorials/django-allauth-tutorial>> [Accessed 17 March 2022].
- GitHub. 2022. *GitHub - django-ckeditor/django-ckeditor: Django admin CKEditor integration..* [online] Available at: <<https://github.com/django-ckeditor/django-ckeditor>> [Accessed 17 March 2022].
- vsupalov.com. 2022. *What Is Gunicorn, and What Does It Do?*. [online] Available at: <<https://vsupalov.com/what-is-gunicorn/>> [Accessed 17 March 2022].
- GitHub. 2022. *GitHub - oauthlib/oauthlib: A generic, spec-compliant, thorough implementation of the OAuth request-signing logic.* [online] Available at: <<https://github.com/oauthlib/oauthlib>> [Accessed 17 March 2022].
- Tutorialspoint.com. 2022. *Python Pillow - Quick Guide*. [online] Available at: <[https://www.tutorialspoint.com/python\\_pillow/python\\_pillow\\_quick\\_guide.htm](https://www.tutorialspoint.com/python_pillow/python_pillow_quick_guide.htm)> [Accessed 17 March 2022].
- Heroku.com. 2022. *About Heroku | Heroku*. [online] Available at: <<https://www.heroku.com/about>> [Accessed 18 March 2022].
- Google Developers. 2022. *Integrating Google Sign-In into your web app | Google Sign-In for Websites | Google Developers*. [online] Available at: <<https://developers.google.com/identity/sign-in/web/sign-in>> [Accessed 18 March 2022].