

Topic: Car Listing: Search Functionality Part 10

Speaker: Udemy Instructor Rathan Kumar | Notebook: Django Project: Car Listing



1. To allow searching based on other fields like MODEL, YEAR, CITY, or BODY_STYLE, update the SEARCH function in CARS/VIEWS.PY

FROM:

```
def search(request):
    cars = Car.objects.order_by('-created_date')

    if 'keyword' in request.GET:
        keyword = request.GET['keyword']

        # checks if keyword is not blank
        if keyword:
            # filters the cars containing keyword
            cars = cars.filter(description__icontains=keyword)

    data = {
        'cars': cars,
    }
    return render(request, 'cars/search.html', data)
```

TO:

```
32 def search(request):
33     cars = Car.objects.order_by('-created_date')
34
35     if 'keyword' in request.GET:
36         keyword = request.GET['keyword']
37
38         # checks if keyword is not blank
39         if keyword:
40             # filters the cars containing keyword
41             cars = cars.filter(description__icontains=keyword)
42
43     if 'model' in request.GET:
44         model = request.GET['model']
45
46         # checks if model is not blank
47         if model:
48             # filters the cars containing keyword
49             cars = cars.filter(model__iexact=model)
50
51     if 'city' in request.GET:
52         city = request.GET['city']
53
54         if city:
55             cars = cars.filter(city__iexact=city)
56
57     if 'year' in request.GET:
58         year = request.GET['year']
59
60         if year:
61             cars = cars.filter(year__iexact=year)
62
63     if 'body_style' in request.GET:
64         body_style = request.GET['body_style']
65
66         if body_style:
67             cars = cars.filter(body_style__iexact=body_style)
68
69     data = {
70         'cars': cars,
71     }
72     return render(request, 'cars/search.html', data)
73
74
```

2. We change the name of our SELECT attributes

FROM:

```

<div class="form-group">
    <select class="form-control search-fields" name="select-make">
        <option selected="true" disabled="disabled">Model</option>
        {% for model in model_search %}
            <!-- calling the list of model values and not a dictionary -->
            <option value="{{model}}">{{model}}</option>
        {% endfor %}
    </select>
</div>
<div class="form-group">
    <select class="form-control search-fields" name="select-location">
        <option selected="true" disabled="disabled">Location</option>
        {% for city in city_search %}
            <option value="{{city}}">{{city}}</option>
        {% endfor %}
    </select>
</div>

```

TO:

```

<select class="form-control search-fields" name="model">
    <option selected="true" disabled="disabled">Model</option>
    {% for model in model_search %}
        <!-- calling the list of model values and not a dictionary -->
        <option value="{{model}}">{{model}}</option>
    {% endfor %}
</select>
</div>
<div class="form-group">
    <select class="form-control search-fields" name="city">
        <option selected="true" disabled="disabled">Location</option>
        {% for city in city_search %}
            <option value="{{city}}">{{city}}</option>
        {% endfor %}
    </select>
</div>

```

3. For the PRICE SLIDER. We use GTE (FOR GREATER THAN OR EQUAL) and LTE (FOR LESS THAN EQUAL)

* take note of the double underscore before the use of LTE or GTE (__gte or __lte)

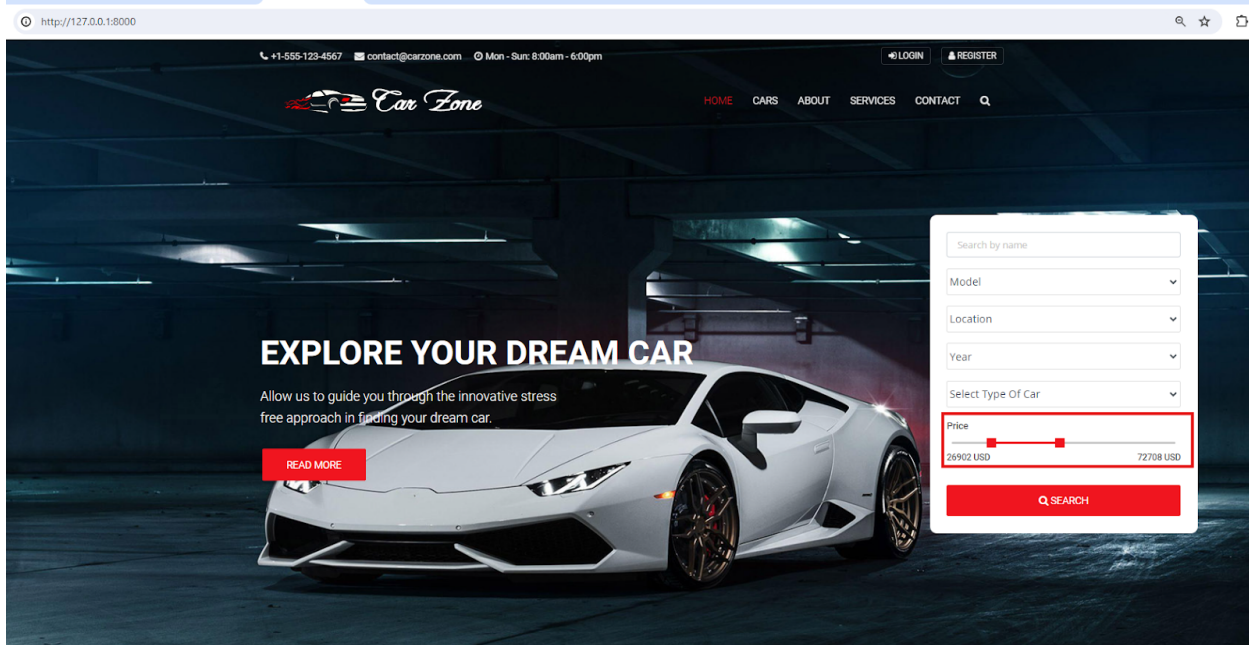
```

if 'min_price' in request.GET:
    min_price = request.GET['min_price']
    max_price = request.GET['max_price']

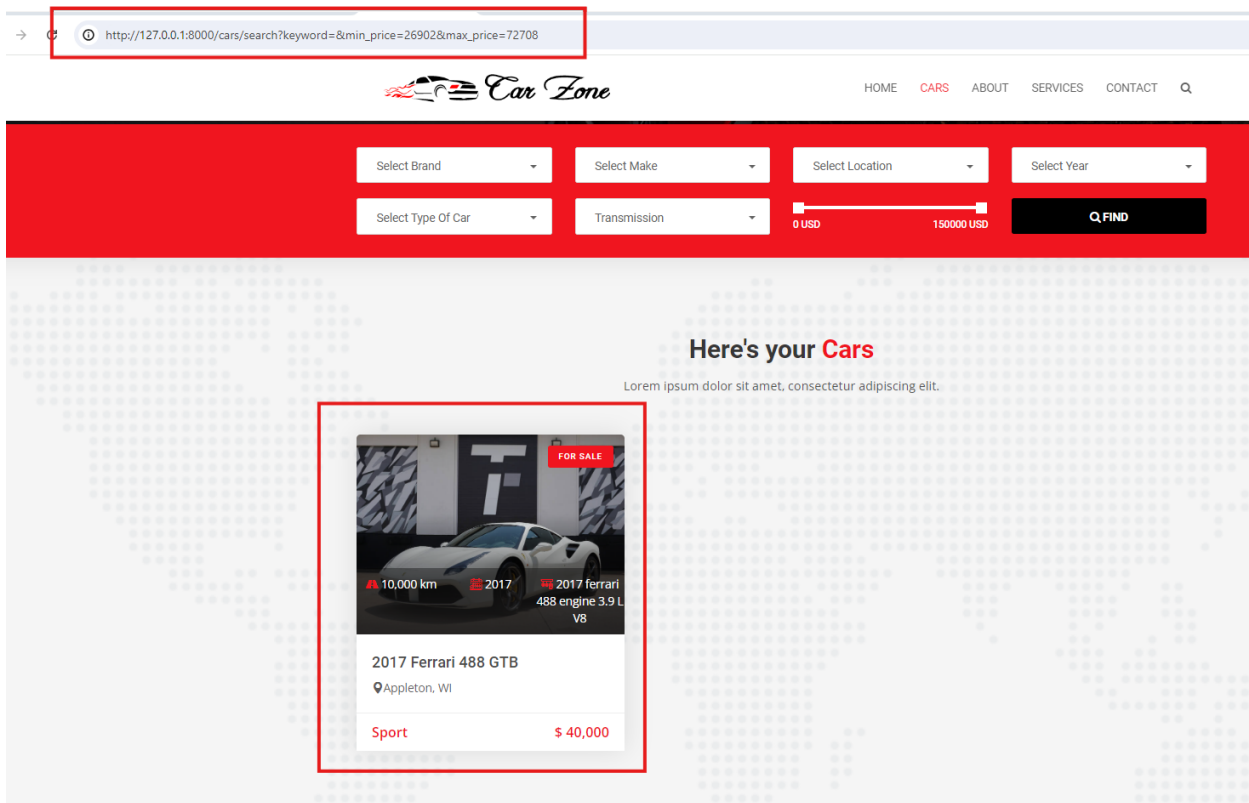
    if max_price:
        # compares price >= for gte and <= for lte
        cars = cars.filter(price__gte=min_price, price__lte=max_price)

```

Run the server again to search using the price slider:



Result should be:



4. We do the same for the SEARCH functionality on the CARS web page. We copy the code from HOME.HTML search functionality into CARS.HTML.

We replace this block with the search block from the HOME.HTML

```
Carzone-Project
EXPLOSER
CARZONE-PROJECT
  templates
  cars
  ...pycache...
  migrations
  ..._int...py
  admin.py
  apps.py
  models.py
  tests.py
  urls.py
  views.py
  carzone
  env
  media
  pages
  ...pycache...
  migrations
  ..._int...py
  admin.py
  apps.py
  models.py
  tests.py
  urls.py
  views.py
  photos
  Resources
  static
  templates
  cars
  car_detail.html
  cars.html
  includes
  footer.html
  navbar.html
  topbar.html
  pages
  about.html
  contact.html
  home.html
  services.html
  base.html
  .gitignore
  carzone_db
  db.sqlite3
  manage.py
  24 <div class="featured-car content-area">
  25 <div class="container">
  26 <div class="row">
  27 <div class="col-lg-8 col-md-12">
  145 </div>
  146 <div class="col-lg-4 col-md-12">
  147 <div class="sidebar-right">
  148 <!-- Advanced search start -->
  149 <div class="widget advanced-search2">
  150 <h3 class="sidebar-title">Search your desire car</h3>
  151 <form method="GET">
  152 <div class="form-group">
  153 <select class="selectpicker search-fields" name="brand">
  154 <option>Brand</option>
  155 <option>Any</option>
  156 <option>New</option>
  157 <option>Semi-New</option>
  158 <option>Damaged</option>
  159 <option>Used</option>
  160 </select>
  161 </div>
  162 <div class="form-group">
  163 <select class="selectpicker search-fields" name="location">
  164 <option>Location</option>
  165 <option>United States</option>
  166 <option>United Kingdom</option>
  167 <option>American Samoa</option>
  168 <option>Belgium</option>
  169 <option>Cameroon</option>
  170 <option>Canada</option>
  171 </select>
  172 </div>
  173 <div class="form-group">
  174 <select class="selectpicker search-fields" name="year">
  175 <option>Year</option>
  176 <option>2016</option>
  177 <option>2017</option>
  178 <option>2018</option>
  179 <option>2019</option>
  180 <option>2020</option>
  181 </select>
  182 </div>
  183 <div class="form-group">
  184 <select class="selectpicker search-fields" name="category">
  185 <option>Category</option>
  186 <option>Luxury Car</option>
  187 <option>Pickup Truck</option>
  188 <option>Minivan</option>
  189 <option>Truck</option>
  190 <option>Sports Car</option>
  191 <option>Wagon</option>
  192 </select>
  193 </div>
  194 <div class="form-group">
  195 <select class="selectpicker search-fields" name="transmission">
  196 <option>Transmission</option>
  197 <option>Automatic</option>
  198 <option>Manual</option>
  199 <option>Tiptronic</option>
  200 </select>
  201 </div>
  202 <div class="range-slider clearfix">
  203 <label>Price</label>
  204 <div data-min="0" data-max="150000" data-min-name="min_price" data-max-name="max_price"
  205 data-unit="USD" class="range-slider-ui ui-slider" aria-disabled="false"></div>
```

New code:

```
Carzone-Project
DEVELOPER
CARZONE-PROJECT
templates > cars > 24 <div class="featured-car content-area">
25 <div class="container">
26 <div class="row">
146 <div class="col-lg-4 col-md-12">
147 <div class="sidebar-right">
148 <!-- Advanced search start -->
149 <div class="widget advanced-search2">
150 <h3 class="sidebar-title">Search your desire car</h3>
151 <!-- Search box 3 start -->
152 <div class="search-box-4">
153 <form action="{% url 'search' %}" method="">
154 <div class="form-group">
155 <input type="text" name="keyword" placeholder="Search by name" class="form-control">
156 </div>
157 <div class="form-group">
158 <select class="form-control search-fields" name="model">
159 <option selected="true" disabled="disabled">Model</option>
160 {% for model in model_search %}
161 <option value="{{model}}">{{model}}</option>
162 <!-- calling the list of model values and not a dictionary -->
163 </select>
164 </div>
165 <div class="form-group">
166 <select class="form-control search-fields" name="city">
167 <option selected="true" disabled="disabled">Location</option>
168 {% for city in city_search %}
169 <option value="{{city}}">{{city}}</option>
170 </select>
171 </div>
172 <div class="form-group">
173 <select class="form-control search-fields" name="year">
174 <option selected="true" disabled="disabled">Year</option>
175 {% for year in year_search %}
176 <option value="{{year}}">{{year}}</option>
177 </select>
178 </div>
179 <div class="form-group">
180 <select class="form-control search-fields" name="body_style">
181 <option selected="true" disabled="disabled">Select Type Of Car</option>
182 {% for type_of_car in body_style_search %}
183 <option value="{{type_of_car}}">{{type_of_car}}</option>
184 </select>
185 </div>
186 <div class="range-slider clearfix">
187 <label>Price</label>
188 <div data-min="0" data-max="150000" data-min-name="min_price"
189 data-max-name="max_price" data-unit="USD" class="range-slider-ui ui-slider"
190 aria-disabled="false"></div>
191 <div class="clearfix"></div>
192 </div>
193 <div class="form-group">
194 <button class="btn btn-block button-theme btn-md">
195 <i class="fa fa-search"></i> Search
196 </button>
197 </div>
198 </form>
199 </div>
200 <!-- Search box 3 end -->
201 </div>
202 </div>
203 </div>
204 </div>
205 </div>
206 </div>
207
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE
WARNINGS:
P: (django-admin) Django-ckeditor bundles CKEditor 4.12.1 which isn't supported anymore and which does have unfixed security issues, see for example https://ckeditor.com/cke4/relase/CKEditor-4.14.0-LTS . You should consider strengt
r (maybe CKEditor 5 respectively django-ckeditor-5 after checking whether the CKEditor 5 license terms work for you) or switch to the non-free CKEditor 4 LTS package. See https://ckeditor.com/ckeditor-4-support/ for more on this. (By
y the django-ckeditor developers and we are not affiliated with CKSource and were not involved in the licensing change, so please refrain from complaining to us. Thanks.)
cars.Cars: (models.Model) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
NOTE: Configure the DEFAULT_AUTO_FIELD setting or the AppConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
> OUTLINE
System check identified 2 issues (0 silenced).
```

5. Update the CARSVIEWS.PY by copying the search code we did on our home page.

```
1 from django.shortcuts import render, get_object_or_404
2 from .models import Car
3 from django.core.paginator import EmptyPage, PageNotAnInteger, Paginator
4
5 # Create your views here.
6
7
8 def cars(request):
9     cars = Car.objects.order_by('-created_date')
10
11     # Shows 3 cars on the page
12     paginator = Paginator(cars, 4)
13     #
14     page = request.GET.get('page')
15     paged_cars = paginator.get_page(page)
16
17     model_search = Car.objects.values_list('model', flat=True).distinct()
18     city_search = Car.objects.values_list('city', flat=True).distinct()
19     year_search = Car.objects.values_list('year', flat=True).distinct()
20     body_style_search = Car.objects.values_list(
21         'body_style', flat=True).distinct()
22
23     data = {
24         'cars': paged_cars,
25         'model_search': model_search,
26         'city_search': city_search,
27         'year_search': year_search,
28         'body style search': body_style_search,
29     }
30     return render(request, 'cars/cars.html', data)
31
```

We copy what we did from the PAGES/VIEWS.PY for HOME.HTML search function.

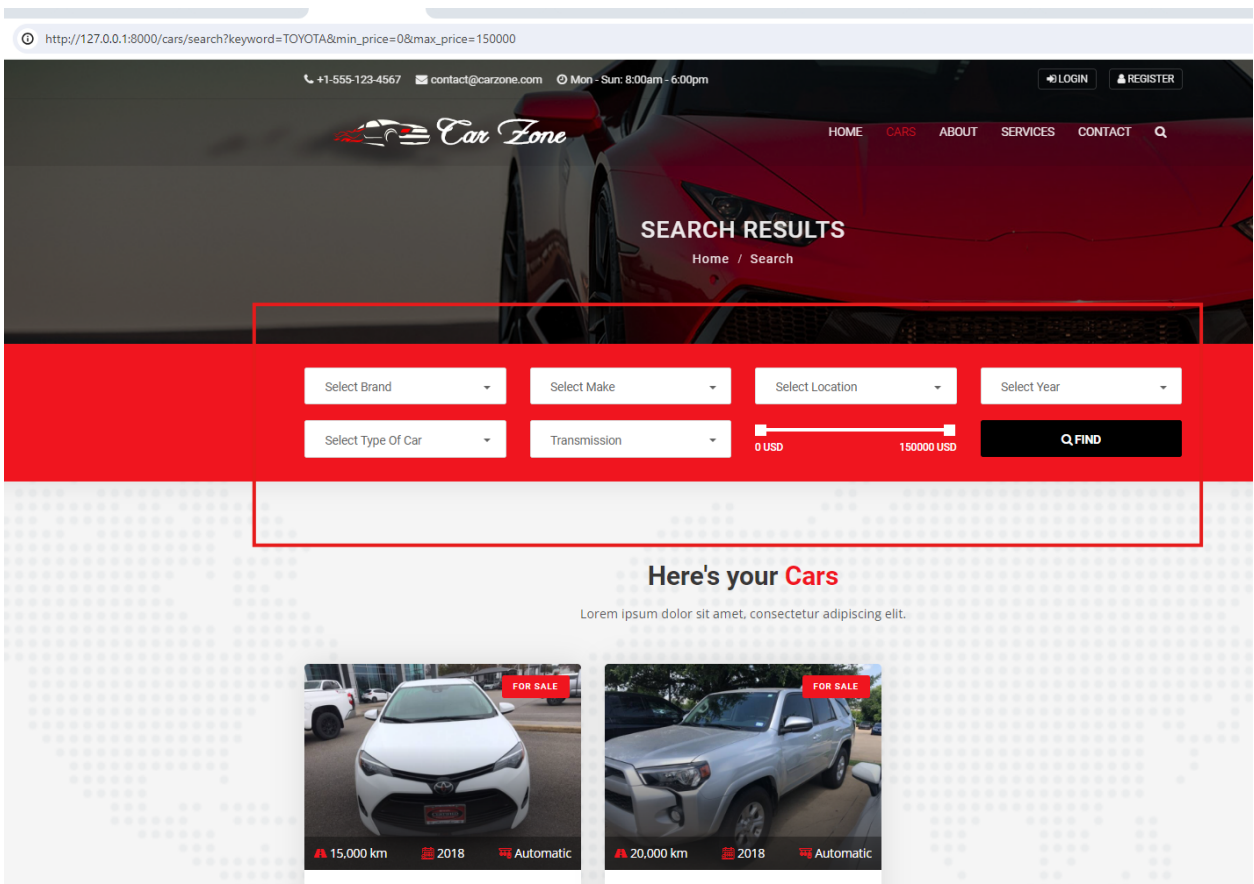
from PAGES/VIEWS.PY

```

1 from django.shortcuts import render
2 from .models import Team
3 from cars.models import Car
4
5
6 # Create your views here.
7 def home(request):
8     # get all the team members
9     teams = Team.objects.all()
10    featured_cars = Car.objects.order_by(
11        '-created_date').filter(is_featured=True)
12    all_cars = Car.objects.order_by('-created_date')
13    # extracts the values of these fields for searching
14    # search_fields = Car.objects.values('model', 'city', 'year', 'body_style')
15
16    # extracts only unique values on models into a list
17    model_search = Car.objects.values_list('model', flat=True).distinct()
18    city_search = Car.objects.values_list('city', flat=True).distinct()
19    year_search = Car.objects.values_list('year', flat=True).distinct()
20    body_style_search = Car.objects.values_list(
21        'body_style', flat=True).distinct()
22
23    data = {
24        'teams': teams,
25        'featured_cars': featured_cars,
26        'all_cars': all_cars,
27        # 'search_fields': search_fields,
28        'model_search': model_search,
29        'city_search': city_search,
30        'year_search': year_search,
31        'body_style_search': body_style_search,
32    }
33
34    return render(request, 'pages/home.html', data)

```

6. We complete the TOP SEARCH functionality



We remove unwanted fields like BRAND. Go to SEARCH.HTML and replace the name:

FROM:

```
<div class="col-lg-3 col-md-6 col-sm-6 col-6">
  <div class="form-group">
    <select class="selectpicker search-fields" name="select-make">
      <option disabled="disabled" selected="selected">Select Make</option>
      <option>BMW</option>
      <option>Honda</option>
      <option>Lamborghini Huracán</option>
      <option>Sports Car</option>
    </select>
  </div>
</div>
<div class="col-lg-3 col-md-6 col-sm-6 col-6">
  <div class="form-group">
    <select class="selectpicker search-fields" name="select-location">
      <option disabled="disabled" selected="selected">Select Location</option>
      <option>United States</option>
      <option>United Kingdom</option>
    </select>
  </div>
</div>
<div class="col-lg-3 col-md-6 col-sm-6 col-6">
  <div class="form-group">
    <select class="selectpicker search-fields" name="select-year">
      <option disabled="disabled" selected="selected">Select Year</option>
      <option>2016</option>
      <option>2017</option>
      <option>2018</option>
      <option>2019</option>
    </select>
  </div>
</div>
```

TO:

```
Carzone-Project
views.py car M
search.html U X
car_detail.html U
cars.html M
views.py pages M
contact.html
services.html
about.html
home.html M
tests.py
topbar.html

CARZONE-PROJECT
> cars
  > pycache
  > migrations
  > __init__.py
  > admin.py
  > apps.py
  > models.py
  > tests.py
  > urls.py
  > views.py
  > carzone
  > env
  > media
  > pages
  > pycache
  > migrations
  > __init__.py
  > admin.py
  > apps.py
  > models.py
  > tests.py
  > urls.py
  > views.py
  > photos
  > Resources
  > static
  > templates
  > cars
  > car_detail.html
  > cars.html
  > search.html
  > topbar.html
  > navbar.html
  > topbar.html
  > pages
  > about.html
  > admin.html
  > contact.html
  > home.html
  > services.html
  > base.html
  > gllgnore
  > carzone_db
  > debugpy
  > manage.py

21 <div class="search-box-3 sb-7 bg-active">
22 <div class="container">
23 <div class="search-area-inner">
24 <div class="search-contents">
25 <form method="GET" action="{% url 'search' %}">
26
27
28 <div class="col-lg-3 col-md-6 col-sm-6 col-6">
29 <div class="form-group">
30 <select class="selectpicker search-field" name="model">
31 <option disabled="disabled" selected="selected">Select Model</option>
32 {% for model in model_search %}
33 <!-- calling the list of model values and not a dictionary -->
34 <option value="{model}">{{model}}</option>
35 {% endfor %}
36 </select>
37 </div>
38 </div>
39 <div class="col-lg-3 col-md-6 col-sm-6 col-6">
40 <div class="form-group">
41 <select class="selectpicker search-fields" name="city">
42 <option disabled="disabled" selected="selected">Select Location</option>
43 {% for city in city_search %}
44 <option value="{city}">{{city}}</option>
45 {% endfor %}
46 </select>
47 </div>
48 </div>
49 <div class="col-lg-3 col-md-6 col-sm-6 col-6">
50 <div class="form-group">
51 <select class="selectpicker search-fields" name="year">
52 <option disabled="disabled" selected="selected">Select Year</option>
53 {% for year in year_search %}
54 <option value="{year}">{{year}}</option>
55 {% endfor %}
56 </select>
57 </div>
58 </div>
59
60 </div>
61 <div class="row">
62 <div class="col-lg-3 col-md-6 col-sm-6 col-6">
63 <div class="form-group">
64 <select class="selectpicker search-fields" name="body_style">
65 <option disabled="disabled" selected="selected">Select Type Of Car</option>
66 {% for type_of_car in body_style_search %}
67 <option value="{type_of_car}">{{type_of_car}}</option>
68 {% endfor %}
69 </select>
70 </div>
71 </div>
72 <div class="col-lg-3 col-md-6 col-sm-6 col-6">
73 <div class="form-group">
```

7. We update the SEARCH function in the CARSVIEWS.PY

```
Carzone-Project
views.py cars M X search.html U car_detail.html U cars.html M views.py pages M contact.html U
cars > views.py > search
43 def search(request):
44     cars = Car.objects.order_by('-created_date')
45
46     model_search = Car.objects.values_list('model', flat=True).distinct()
47     city_search = Car.objects.values_list('city', flat=True).distinct()
48     year_search = Car.objects.values_list('year', flat=True).distinct()
49     body_style_search = Car.objects.values_list(
50         'body_style', flat=True).distinct()
51     transmission_search = Car.objects.values_list(
52         'transmission', flat=True).distinct()
53
54     if 'keyword' in request.GET:
55         keyword = request.GET['keyword']
56
57         # checks if keyword is not blank
58         if keyword:
59             # filters the cars containing keyword
60             cars = cars.filter(description__icontains=keyword)
61
62     if 'model' in request.GET:
63         model = request.GET['model']
64
65         # checks if model is not blank
66         if model:
67             # filters the cars containing keyword
68             cars = cars.filter(model__iexact=model)
69
70     if 'transmission' in request.GET:
71         transmission = request.GET['transmission']
72
73         if transmission:
74             cars = cars.filter(transmission__iexact=transmission)
75
76     if 'city' in request.GET:
77         city = request.GET['city']
78
79         if city:
80             cars = cars.filter(city__iexact=city)
81
82     if 'year' in request.GET:
83         year = request.GET['year']
84
85         if year:
86             cars = cars.filter(year__iexact=year)
87
88     if 'body_style' in request.GET:
89         body_style = request.GET['body_style']
90
91         if body_style:
92             cars = cars.filter(body_style__iexact=body_style)
93
94     if 'min_price' in request.GET:
95         min_price = request.GET['min_price']
96         max_price = request.GET['max_price']
97
98         if max_price:
99             # compares price >= for double underscore gte and <= for double underscore lte
100             cars = cars.filter(price__gte=min_price, price__lte=max_price)
101
102     data = {
103         'cars': cars,
104         'model_search': model_search,
105         'city_search': city_search,
106         'year_search': year_search,
107         'body_style_search': body_style_search,
108         'transmission_search': transmission_search,
109     }
110     return render(request, 'cars/search.html', data)
111
```

```
cars.html M
search.html U
includes
  footer.html
  navbar.html
  topbar.html
pages
  about.html
  contact.html
  home.html M
  services.html M
  base.html M
.gibgnore
carzone_db M
db.sqlite3
manage.py
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
```

8. Run the server.

The screenshot shows a web browser window with the address bar containing the URL: `http://127.0.0.1:8000/cars/search?transmission=Manual&min_price=0&max_price=150000`. The website header features the 'Car Zone' logo and navigation links for HOME, CARS, ABOUT, SERVICES, and CONTACT. Below the header is a search bar with the text 'Home / Search'. A prominent red search filter bar contains several dropdown menus: 'Select Model', 'Select Location', 'Select Year', 'Select Type Of Car', and 'Transmission'. A price range slider is set from 0 USD to 150,000 USD, and a 'Q.FIND' button is located to the right. The main content area is titled 'Here's your Cars' and includes a placeholder text: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit.' A search result card for a '2017 Ferrari 488 GTB' is highlighted with a red border. The card features a 'FOR SALE' badge, a car image, and details: 10,000 km, 2017, Manual transmission, located in Appleton, WI, with a price of \$40,000 and a 'Sport' category label.