

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

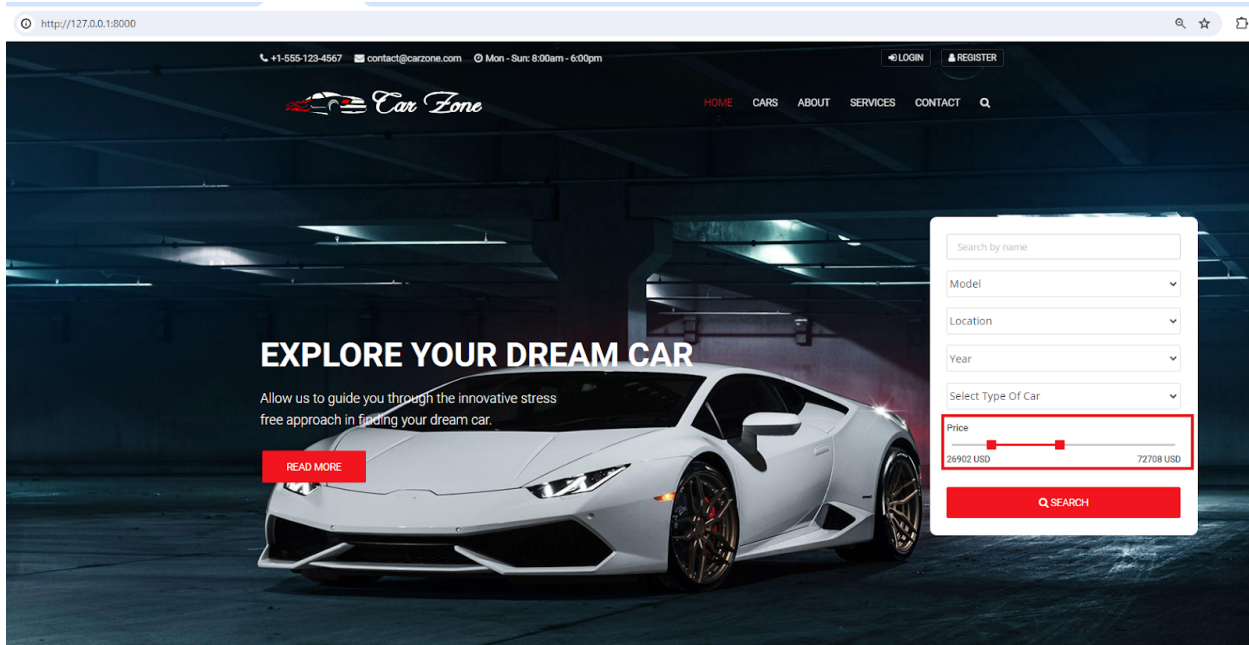
```

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

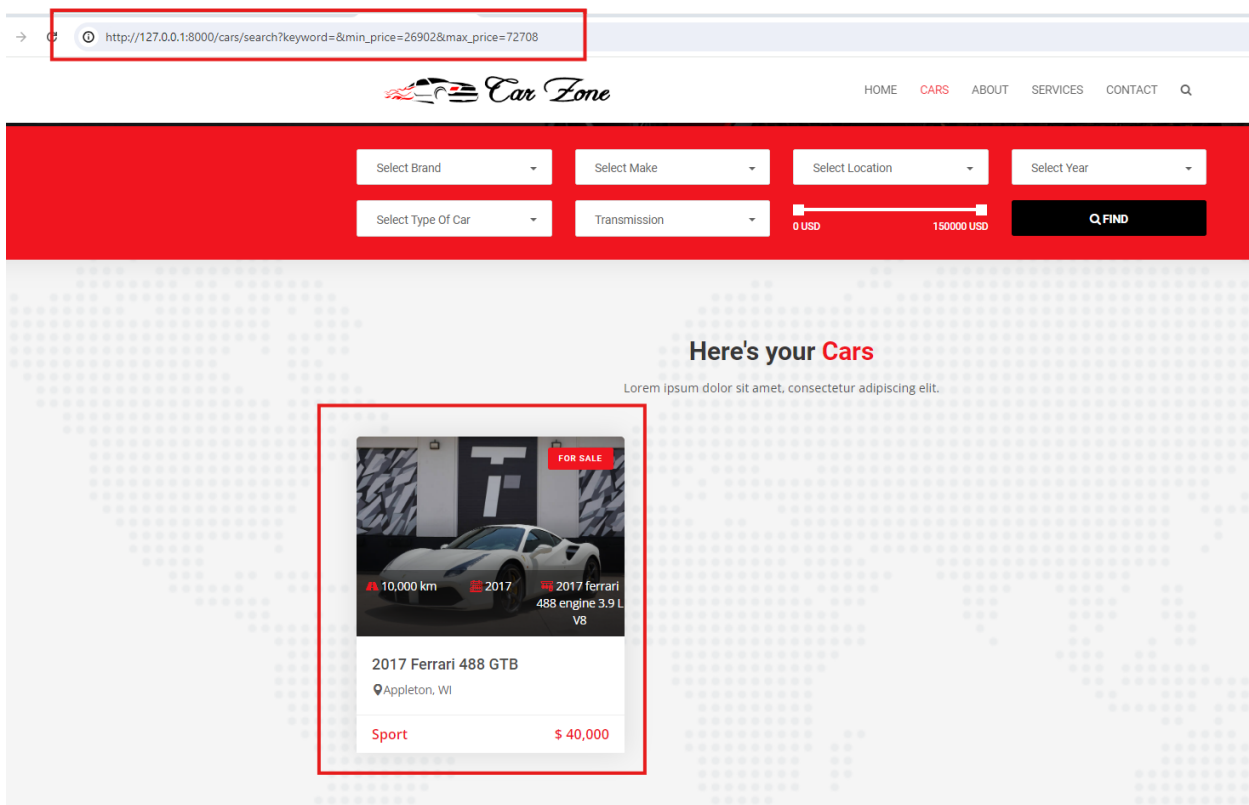
    if max_price:
        # compares price >= for gte and <= for lte
        cars = cars.filter(price__gte=min_price, price__lte=max_price)

```

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

```
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 class="form-group">
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:

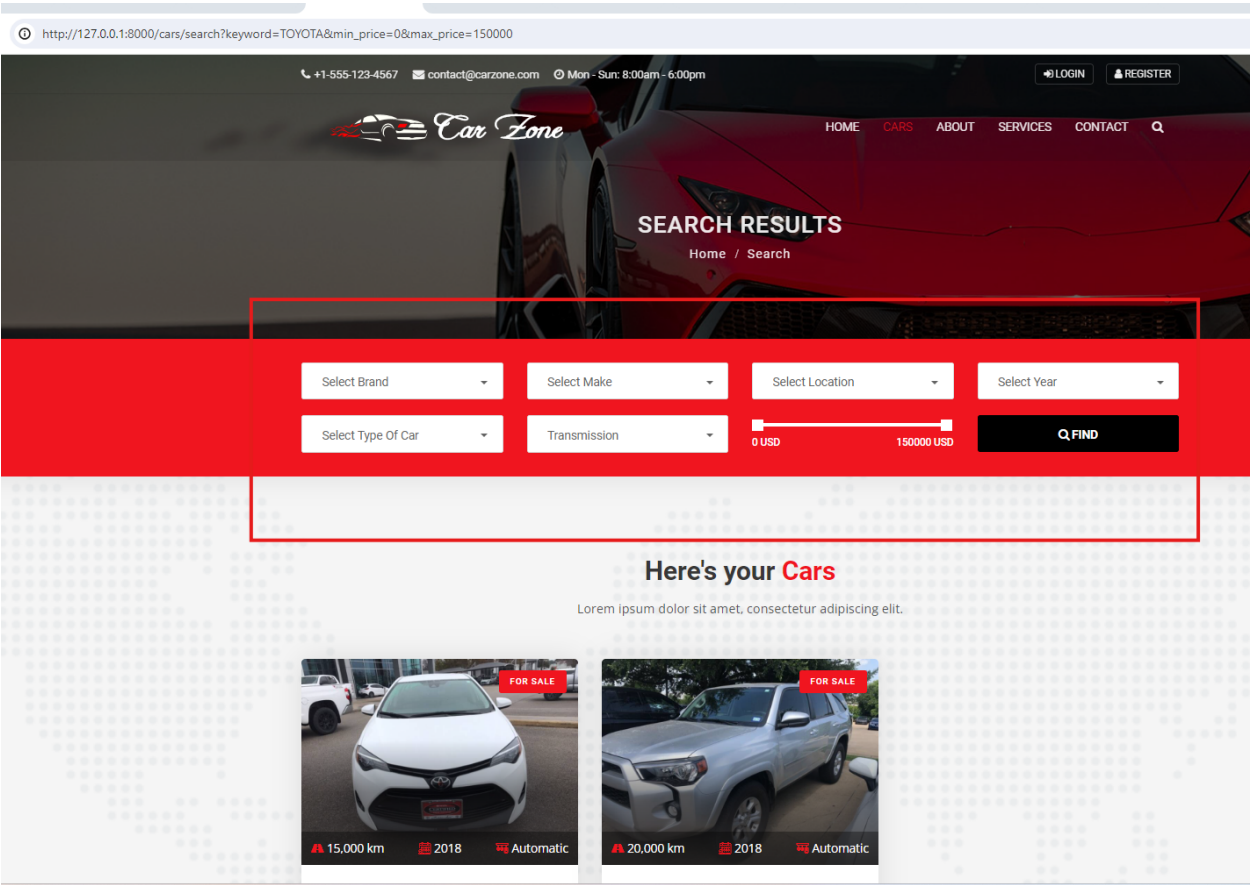

```
1 from django.shortcuts import render, get_object_or_404
2 from .models import Car
3 from django.core.paginator import EmptyPage, PageNotAnInteger, Paginator
4
5 # Create your views here.
6
7
8 def cars(request):
9     cars = Car.objects.order_by('-created_date')
10
11     # Shows 3 cars on the page
12     paginator = Paginator(cars, 4)
13
14     #
15     page = request.GET.get('page')
16     paged_cars = paginator.get_page(page)
17
18     model_search = Car.objects.values_list('model', flat=True).distinct()
19     city_search = Car.objects.values_list('city', flat=True).distinct()
20     year_search = Car.objects.values_list('year', flat=True).distinct()
21     body_style_search = Car.objects.values_list('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)
```

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:

```
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           <div class="col-lg-3 col-md-6 col-sm-6 col-6">
28             <div class="form-group">
29               <select class="selectpicker search-fields" name="model">
30                 <option disabled="disabled" selected="selected">Select Model</option>
31                 <!-- calling the list of model values and not a dictionary -->
32                 <{% for model in model_search %}>
33                   <option value="{{model}}">{{model}}</option>
34                 <{% endfor %}>
35               </select>
36             </div>
37           </div>
38
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
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
73           <div class="col-lg-3 col-md-6 col-sm-6 col-6">
74             <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 serv

CARZONE-PROJECT

cars > views.py > search

cars

views.py

carzone

env

media

pages

photos

Resources

static

templates

cars

car_detail.html

cars.html

search.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

43

def search(request):

44

cars = Car.objects.order_by('-created_date')

45

model_search = Car.objects.values_list('model', flat=True).distinct()

46

city_search = Car.objects.values_list('city', flat=True).distinct()

47

year_search = Car.objects.values_list('year', flat=True).distinct()

48

body_style_search = Car.objects.values_list('body_style', flat=True).distinct()

49

transmission_search = Car.objects.values_list('transmission', flat=True).distinct()

50

if 'keyword' in request.GET:

51

keyword = request.GET['keyword']

52

checks if keyword is not blank

53

if keyword:

54

filters the cars containing keyword

55

cars = cars.filter(description__icontains=keyword)

56

if 'model' in request.GET:

57

model = request.GET['model']

58

checks if model is not blank

59

if model:

60

filters the cars containing keyword

61

cars = cars.filter(model__iexact=model)

62

if 'transmission' in request.GET:

63

transmission = request.GET['transmission']

64

if transmission:

65

cars = cars.filter(transmission__iexact=transmission)

66

if 'city' in request.GET:

67

city = request.GET['city']

68

if city:

69

cars = cars.filter(city__iexact=city)

70

if 'year' in request.GET:

71

year = request.GET['year']

72

if year:

73

cars = cars.filter(year__iexact=year)

74

if 'body_style' in request.GET:

75

body_style = request.GET['body_style']

76

if body_style:

77

cars = cars.filter(body_style__iexact=body_style)

78

if 'min_price' in request.GET:

79

min_price = request.GET['min_price']

80

max_price = request.GET['max_price']

81

compares price >= for double underscore gte and <= for double underscore lte

82

cars = cars.filter(price__gte=min_price, price__lte=max_price)

83

data = {

84

'cars': cars,

85

'model_search': model_search,

86

'city_search': city_search,

87

'year_search': year_search,

88

'body_style_search': body_style_search,

89

'transmission_search': transmission_search,

90

}

91

return render(request, 'cars/search.html', data)

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

carzone.html

search.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

93

cars = cars.filter(body_style__iexact=body_style)

94

if 'min_price' in request.GET:

95

min_price = request.GET['min_price']

96

max_price = request.GET['max_price']

97

compares price >= for double underscore gte and <= for double underscore lte

98

cars = cars.filter(price__gte=min_price, price__lte=max_price)

99

data = {

100

'cars': cars,

101

'model_search': model_search,

102

'city_search': city_search,

103

'year_search': year_search,

104

'body_style_search': body_style_search,

105

'transmission_search': transmission_search,

106

}

107

return render(request, 'cars/search.html', data)

108

109

110

111

8. Run the server.

