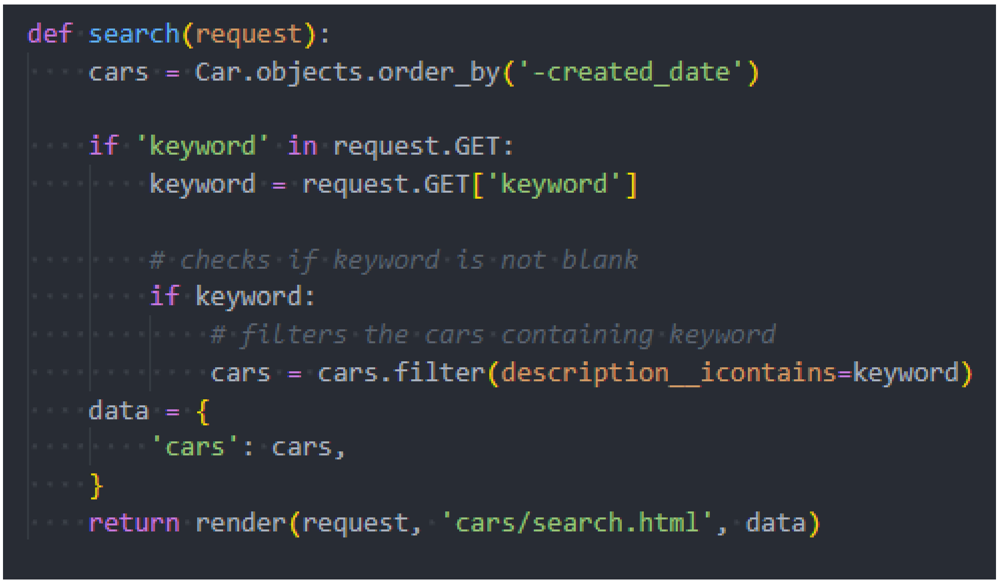
Topic: Car Listing: Search Functionality Part 10

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

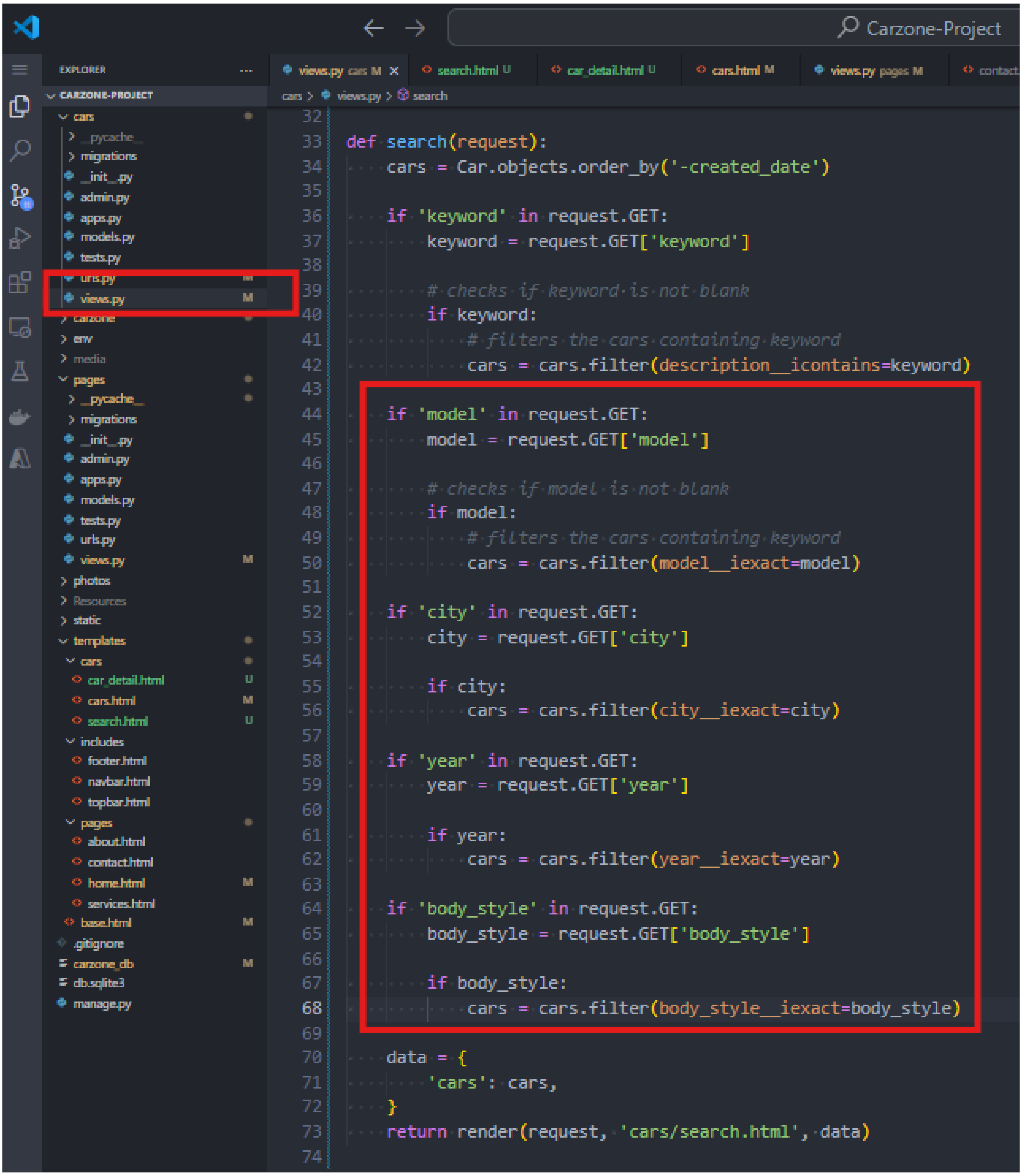
1. To allow searching based on other fields like MODEL, YEAR, CITY, or BODY_STYLE, update the SEARCH function in CARS\VIEWS.PY

FROM:



```python
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)
```

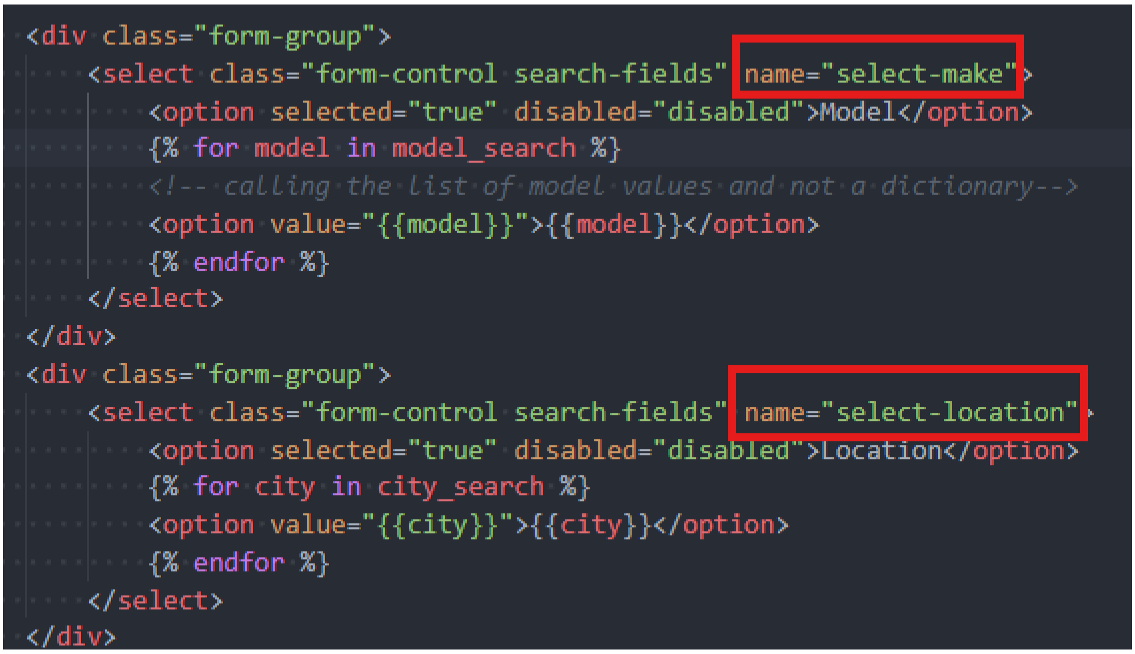
TO:



```python

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)

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

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

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

        if city:
            cars = cars.filter(city__iexact=city)

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

        if year:
            cars = cars.filter(year__iexact=year)

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

        if body_style:
            cars = cars.filter(body_style__iexact=body_style)

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

```

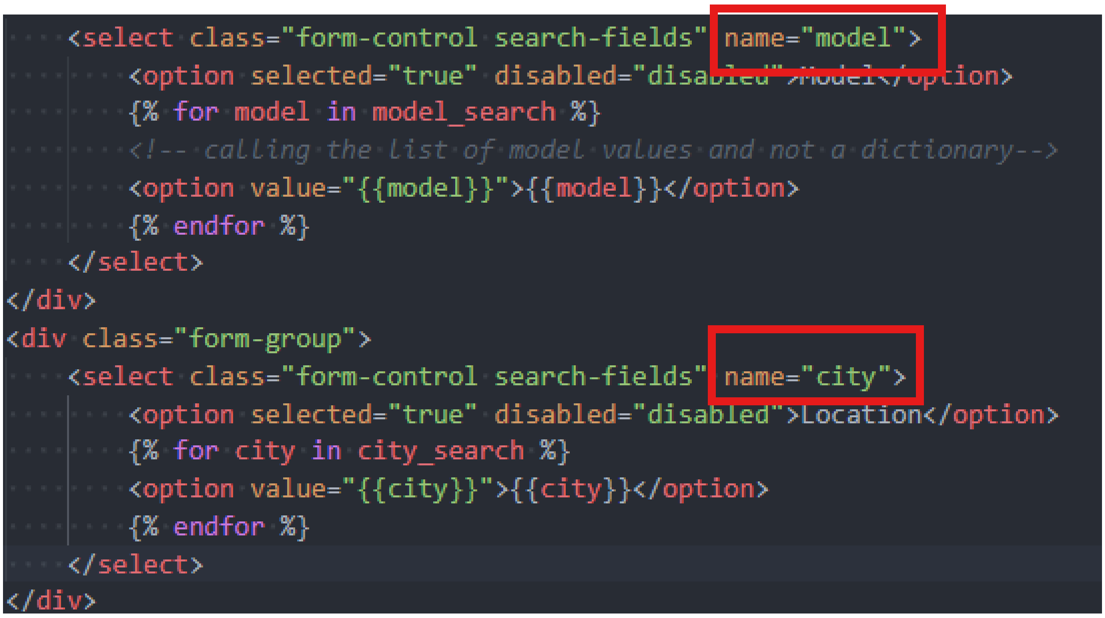
2. We change the name of our SELECT attributes

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)

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

New code:

5. Update the CARS\VIEWS.PY by copying the search code we did on our home page.

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

from PAGES\VIEWS.PY

6. We complete the TOP SEARCH functionality

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

FROM:

TO:

7. We update the SEARCH function in the CARS\VIEWS.PY

8. Run the server.