

# BC Electric Cars

Building upon UI design of given bcre.com, bcec.com for web in e-cars improved searching, enhanced listing, enlightened navbar menu, incorporated a websitemap, and included some more inform pages.

bcec.com [contact@bcec.com](mailto:contact@bcec.com) (852)5555-5555

## BC ELECTRIC CARS

### E-Cars Searching Just Got So Easy

Keyword (Origin, Doors, Seats, ...)

Reset Both Brand and Model to a null criterion

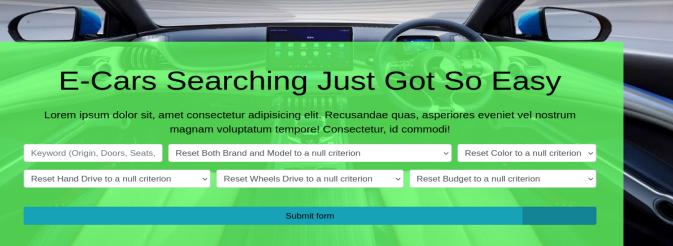
Reset Color to a null criterion

Reset Hard Drive to a null criterion

Reset Wheels Drive to a null criterion

Reset Budget to a null criterion

Submit form



bcec.com [contact@bcec.com](mailto:contact@bcec.com) (852)5555-5555

## BC ELECTRIC CARS

- HOME
- ABOUT
- HISTORY
- VISION
- GOALS
- WORK
- FEATURED LISTINGS
- WEBSITEMAP
- LOGIN
- REGISTER
- ORGANIZATION
- TEAM
- MOST VALUED BROKER(S)
- JOIN US
- CONTACT US
- DISCLAIMER
- PRIVACY DATA POLICY
- PRO COMPETITION POLICY
- FAIR TRADE POLICY
- ENVIRONMENT SOCIETY GOVERNANCE POLICY
- OCCUPATIONAL HEALTH & SAFETY POLICY
- ANTI MONEY LAUNDERING POLICY
- ANTI-BRIBERY-CORRUPTION POLICY
- ADVERSE WEATHER ARRANGEMENTS

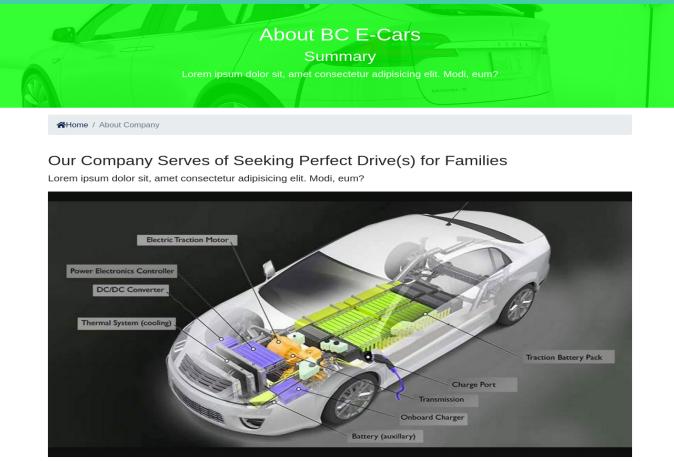
bcec.com [contact@bcec.com](mailto:contact@bcec.com) (852)5555-5555

bcec.com [contact@bcec.com](mailto:contact@bcec.com) (852)5555-5555

## BC ELECTRIC CARS

### About BC E-Cars Summary

Our Company Serves of Seeking Perfect Drive(s) for Families



### 3 Latest E-Cars Listings

Model	Color	Price
Volvo EX30	Yellow	HK\$ 1,200,000
Volkswagen ID.7	Blue	HK\$ 700,000
Tesla Model Y	Blue	HK\$ 1,300,000

More Info (Listed Since: 1 day, 18 hours)

More Info (Listed Since: 1 day, 19 hours)

More Info (Listed Since: 1 day, 19 hours)

3 Latest E-Cars as shown

Origin: Sweden Color: Yellow  
Doors: 4 Seats: 4  
Hand Drive: 4 Wheels Drive: 4  
Mileage: 7000.0 Range: 700.0

Origin: Germany Color: Blue  
Doors: 4 Seats: 5  
Hand Drive: 4 Wheels Drive: 5  
Mileage: 7000.0 Range: 700.0

Origin: USA Color: Blue  
Doors: 2 Seats: 2  
Hand Drive: 2 Wheels Drive: 2  
Mileage: 80000.0 Range: 800.0

E-Car Broker: Jenny [12333-3333](tel:12333-3333) [jenny@bcec.com](mailto:jenny@bcec.com)

E-Car Broker: Mark [7777-7777](tel:7777-7777) [mark@bcec.com](mailto:mark@bcec.com)

E-Car Broker: Kyle [4444-4444](tel:4444-4444) [kyle@bcec.com](mailto:kyle@bcec.com)

## BC E-Cars' Organization Summary

O-chart For The Company

BC E-Cars' Orgchart

```
graph TD; Director[Director Broad President Charles] --> Management[Management Broad CEO William]; Director --> Executing[Executing Team Leader Edward]; Management --> Executing; Management --> Advisory[Advisory Broad Coordinator Patrick]; Executing --> Jenny[Consultant Broad Chairhead Jenny]; Executing --> Kyle[Consultant Broad Chairhead Kyle]; Executing --> Mark[Consultant Broad Chairhead Mark];
```

Copyright © 2024 BC Electric Cars

Home About History Vision Goals Work Featured Listings Contact Us Disclaimer

Privacy Data Policy Pro-Competition Policy Fair Trade Policy Environment Society Governance Policy Occupational Health & Safety Policy Anti-Money-Laundering Policy Anti-Bribery-Corruption Policy Adverse Weather Arrangements

## Welcome You Onto The Site Board

Copyright © 2024 BC Electric Cars

# UX Design – Realisticity

Some improvements on bcre website can be relied upon UX Design on the search boxes.

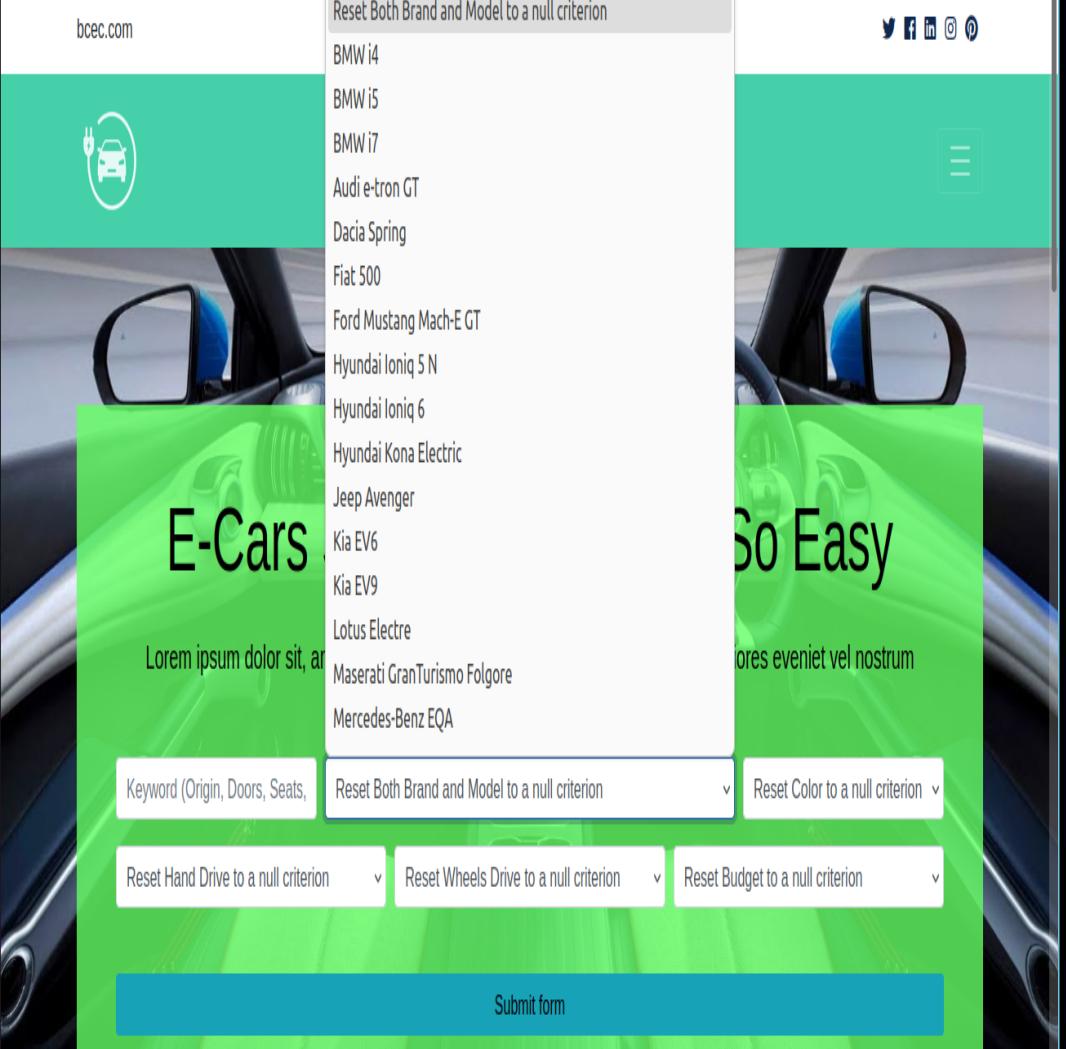
UX can be increased by offer of more realisticity senses to users in the options' choices.

Two things can be done on the search boxes:

1. combinations of search choices
2. real search choices themselves

Let see the bcec website search box first: P.T.O.

# search boxes and search fields



The screenshot shows a web page for "E-Cars" featuring a sidebar with a car icon and a list of electric vehicle models. The main area displays a green car image with the text "So Easy" overlaid. Below the image are several search filters and a "Submit form" button.

Search filters include dropdowns for "Keyword (Origin, Doors, Seats," "Reset Both Brand and Model to a null criterion," "Reset Color to a null criterion," "Reset Hand Drive to a null criterion," "Reset Wheels Drive to a null criterion," and "Reset Budget to a null criterion."

The sidebar lists the following car models:

- BMW i4
- BMW i5
- BMW i7
- Audi e-tron GT
- Dacia Spring
- Fiat 500
- Ford Mustang Mach-E GT
- Hyundai Ioniq 5 N
- Hyundai Ioniq 6
- Hyundai Kona Electric
- Jeep Avenger
- Kia EV6
- Kia EV9
- Lotus Electre
- Maserati GranTurismo Folgore
- Mercedes-Benz EQA

At the bottom of the sidebar, there is placeholder text: "Lorem ipsum dolor sit, amet."

On the right side of the slide, there is a code snippet for a Python model named `Listing`:

```
5 class Listing(models.Model):
6     broker = models.ForeignKey(Broker, on_delete=models.DO_NOTHING)
7     brand = models.CharField(max_length=50)
8     model = models.CharField(max_length=100)
9     origin = models.CharField(max_length=50)
10    color = models.CharField(max_length=50)
11    doors = models.IntegerField()
12    seats = models.IntegerField()
13    milage = models.DecimalField(max_digits=6, decimal_places=1)
14    range = models.DecimalField(max_digits=4, decimal_places=1)
15    hand_drive = models.CharField(max_length=50)
16    wheels_drive = models.CharField(max_length=50)
17    price = models.IntegerField()
18    description = models.TextField(blank=True)
19    list_date = models.DateTimeField(auto_now_add=True)
20    is_published = models.BooleanField(default=True)
21    photo_main = models.ImageField(upload_to='photos/%Y/%m/%d/')
22    photo_1 = models.ImageField(upload_to='photos/%Y/%m/%d/', blank=True)
23    photo_2 = models.ImageField(upload_to='photos/%Y/%m/%d/', blank=True)
24    photo_3 = models.ImageField(upload_to='photos/%Y/%m/%d/', blank=True)
25    photo_4 = models.ImageField(upload_to='photos/%Y/%m/%d/', blank=True)
26    photo_5 = models.ImageField(upload_to='photos/%Y/%m/%d/', blank=True)
27    photo_6 = models.ImageField(upload_to='photos/%Y/%m/%d/', blank=True)
28
29    def __str__(self):
30        return self.brand + ' ' + self.model
```

# Combinations of Search Choices / Fields

Some relatively major search fields / choices in this case such as brand and model need to be combined to restrict from unrealistic combinations to drive out users' possible confusions in selecting options to keep up their rigor.

The screenshot shows a web page for "E-Cars" (bcec.com) featuring a search interface. A dropdown menu is open, listing various car models under the heading "Reset Both Brand and Model to a null criterion". The listed models include:

- BMW i4
- BMW i5
- BMW i7
- Audi e-tron GT
- Dacia Spring
- Fiat 500
- Ford Mustang Mach-E GT
- Hyundai Ioniq 5 N
- Hyundai Ioniq 6
- Hyundai Kona Electric
- Jeep Avenger
- Kia EV6
- Kia EV9
- Lotus Electre
- Maserati GranTurismo Folgore
- Mercedes-Benz EQA

The page also features other search filters like "Keyword (Origin, Doors, Seats, ...)", "Reset Hand Drive to a null criterion", "Reset Wheels Drive to a null criterion", "Reset Color to a null criterion", "Reset Budget to a null criterion", and a "Submit form" button.

# Combinations of Search Choices / Fields

Other relatively minor search fields / choices in this case such as color and budget can be individualized as freedom of preferences allowing for unrealistic combinations for example below a budget not a color available or accessible retaining user exploration joy.

The screenshot shows a web-based search interface for electric cars. At the top, there's a logo of a car with a plug and the text "BC ELECTRIC CARS". Below it, the main heading reads "E-Cars Searching Just Got Easier". A placeholder text "Lorem ipsum dolor sit, amet consectetur adipisicing elit. Recusandae quia magnam voluptatum tempore! Consectetur, id co" is visible. There are several search filters: "Keyword (Origin, Doors, Seats, Model)" and "Reset Both Brand and Model to a null criterion". Below these are dropdown menus for "Reset Hand Drive to a null criterion" and "Reset Wheels Drive to a null criterion". On the right, a dropdown menu for "Reset Budget to a null criterion" is open, listing various price points in HKD: HK\$200,000, HK\$400,000, HK\$600,000, HK\$800,000, HK\$1,000,000, HK\$1,200,000, HK\$1,400,000, HK\$1,600,000, HK\$1,800,000, HK\$2,000,000, and HK\$2,200,000. A large blue button at the bottom says "Submit form".

Reset Budget to a null criterion

- HK\$200,000
- HK\$400,000
- HK\$600,000
- HK\$800,000
- HK\$1,000,000
- HK\$1,200,000
- HK\$1,400,000
- HK\$1,600,000
- HK\$1,800,000
- HK\$2,000,000
- HK\$2,200,000

# Real Search Choices of Fields

Real Search fields choices can be derived from db data reflecting true options for potential customers, increasing informative values in selection courses.

The screenshot shows a web application interface for searching electric cars. At the top, there is a navigation bar with a car icon, the text "BC ELECTRIC CARS", and a menu icon. Below the navigation bar, the main title "E-Cars Searching Just Got So Easy" is displayed. A placeholder text "Lorem ipsum dolor sit, amet consectetur adipisicing elit. Recusandae quas, asperiores eveniet vel nostrum magnam voluptatum tempore! Consectetur, id commodi!" follows. The search form includes fields for "Keyword (Origin, Doors, Seats, ...)" and dropdown menus for "Reset Both Brand and Model to a null criterion", "Reset Hand Drive to a null criterion", "Reset Wheels Drive to a null criterion", and "Reset Budget to a null criterion". A "Submit form" button is located at the bottom of the search area. To the right of the search form, a dropdown menu for "Reset Color to a null criterion" is open, showing a list of color options: Silvery, Black, Yellow, Red, White, Orange, Blue, and Grey. The background of the page features a blurred image of a car interior.

# How can we achieve those ideals?

See my share of python / django coding tips, p.t.o.

# to share django / python coding tips: search field color



listings > 🗃 choices.py > ...

```
3     listings = Listing.objects.all().filter(is_published=True)
7     colors_set = {listing.color for listing in listings}
8     color_choices = {'Reset Color to a null criterion': ''} | {color: color for color in colors_set}
```

templates > pages > index.html

```
40     <div class="col-md-3 mb-3">
41         <label class="sr-only" for="color">Colour</label>
42         <select name="color" class="form-control" id="color">
43             {% for key, value in color_choices.items %}
44                 <option value="{{value}}>{{key}}</option>
45             {% endfor %}
46         </select>
47     </div>
```

1: set of colors from db 2:dictionary 3:add reset null to dict head 4:options, dynamic from db

# to share django / python coding tips: search field hand\_drive



BC ELECTRIC CARS



## E-Cars Searching Just Got So Easy

Lore ipsum dolor sit, amet consectetur adipisicing elit. Recusandae quas, asperiores eveniet vel nostrum magnam voluptatum tempore! Consectetur, id commodi!

Keyword (Origin, Doors, Seats, ...)

Reset Both Brand and Model to a null criterion

Reset Hand Drive to a null criterion

Reset Hand Drive to a null criterion

Left-hand-drive

Right-hand-drive

Reset Wheels Drive to a null criterion

Reset Color to a null criterion

Reset Budget to a null criterion

Submit form

listings > choices.py > ...

```
3     listings = Listing.objects.all().filter(is_published=True)
14    hand_drives_set = {listing.hand_drive for listing in listings}
15    hand_drive_choices = {'Reset Hand Drive to a null criterion': ''} | __
16    {hand_drive + '-hand-drive': hand_drive for hand_drive in hand_drives_set}
```

templates > pages > index.html

```
50    <div class="form-row">
51      <div class="col-md-4 mb-3">
52        <label class="sr-only" for="hand_drive">Hand Drive</label>
53        <select name="hand_drive" class="form-control" id="hand_drive">
54          {% for key, value in hand_drive_choices.items %}<br>
55            <option value="{{value}}">{{key}}</option>
56          {% endfor %}
57        </select>
58      </div>
```

1: set of hand\_drives from db 2:dictionary 3:add reset null to dict head 4:options, dynamic from db

to share django / python coding tips: search field wheels drive



listings > choices.py > ...

```
3     listings = Listing.objects.all().filter(is_published=True)
18    wheels_drives_set = {listing.wheels_drive for listing in listings}
19    wheels_drive_choices = {'Reset Wheels Drive to a null criterion': ''} |
20    {wheels_drive + '-wheels-drive': wheels_drive for wheels_drive in wheels_drives_set}
```

templates > pages > index.html

```
59    <div class="col-md-4 mb-3">
60      <label class="sr-only" for="wheels_drive">Wheels Drive</label>
61      <select name="wheels_drive" class="form-control" id="wheels_drive">
62        {% for key, value in wheels_drive_choices.items %}
63          <option value="{{value}}>{{key}}</option>
64        {% endfor %}
65      </select>
66    </div>
```

1: set of wheels\_drives from db 2:dictionary 3:add reset null to dict head 4:options, dynamic from db

# to share django / python coding tips: search field budget



listings > choices.py > ...

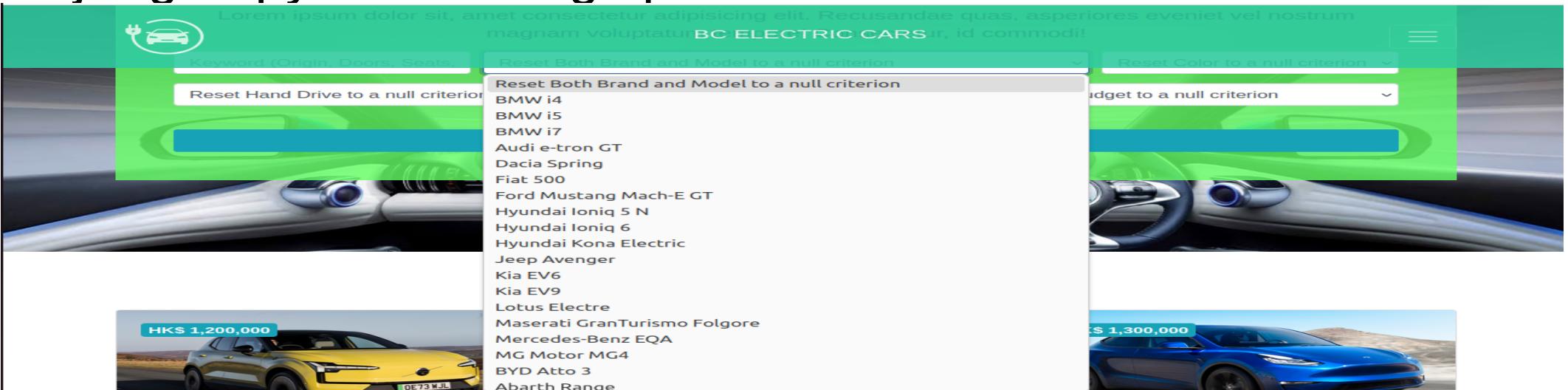
```
3     listings = Listing.objects.all().filter(is_published=True)
10    max_price = max({listing.price for listing in listings})
11    grade = 200000
12    budget_choices = {'Reset Budget to a null criterion': '0'} |_
13    {'HK$'+f'{g:,}': str(g) for g in range(grade, max_price + grade, grade)}
```

templates > pages > index.html

```
67
68
69
70
71
72
73
74      <div class="col-md-4 mb-3">
          <label class="sr-only" for="budget">Budget</label>
          <select name="budget" class="form-control" id="budget">
              {% for key, value in budget_choices.items %}
                  <option value="{{value}}>{{key}}</option>
              {% endfor %}
          </select>
      </div>
```

1: max of set of prices from db 2:grade 3: grades of range() 4: dictionary 3:add reset null to dict head 4:options, dynamic from db

# django / python coding tips: combination field brand & model



listings > choices.py > ...

```
3     listings = Listing.objects.all().filter(is_published=True)
5     brand_model_choices = {'Reset Both Brand and Model to a null criterion': ''} |
6     {listing.brand + ' ~ ' + listing.model: listing.brand + '~' + listing.model for listing in listings}
```

templates > pages > index.html

```
32     <div class="col-md-6 mb-3">
33         <label class="sr-only" for="brand_model">Brand Model</label>
34         <select name="brand_model" class="form-control" id="brand_model">
35             {% for key, value in brand_model_choices.items %}
36                 <option value="{{value}}>{{key}}</option>
37             {% endfor %}
38         </select>
39     </div>
```

1:dictionary of brand+'~'+model : brand+'~'+model 2:add reset null to dict head 3:options, dynamic from db

# django / python coding tips: the search code

listings > views.py > search

```
35 def search(request):
36     queryset_list = Listing.objects.order_by('-list_date').filter(is_published=True)
37     if 'keywords' in request.GET:
38         keywords = request.GET['keywords']
39         if keywords:
40             queryset_list = queryset_list.filter(description__icontains=keywords)
41     if 'brand_model' in request.GET:
42         brand_model = request.GET['brand_model']
43         if brand_model:
44             i = brand_model.index('~')
45             queryset_list = queryset_list.filter(brand__iexact=brand_model[:i]).filter(model__iexact=brand_model[i+1:])
46     if 'color' in request.GET:
47         color = request.GET['color']
48         if color:
49             queryset_list = queryset_list.filter(color__iexact=color)
50     if 'budget' in request.GET:
51         budget = int(request.GET['budget'])
52         if budget:
53             queryset_list = queryset_list.filter(price__lte=budget)
54     if 'hand_drive' in request.GET:
55         hand_drive = request.GET['hand_drive']
56         if hand_drive:
57             queryset_list = queryset_list.filter(hand_drive__iexact=hand_drive)
58     if 'wheels_drive' in request.GET:
59         wheels_drive = request.GET['wheels_drive']
60         if wheels_drive:
61             queryset_list = queryset_list.filter(wheels_drive__iexact=wheels_drive)
```

1.quest for search field key in json  
2.read in search field value in json  
3.screen out null criterion search passing wholly  
4.search non-null criterion  
5.pass to repeat till all search fields  
6.for brand & model: double searches / filters in succession

# Conclusion

UX is an important niche of software design as ultimate users are human. It much increases the value of use of the software.

Thank you for your participation.