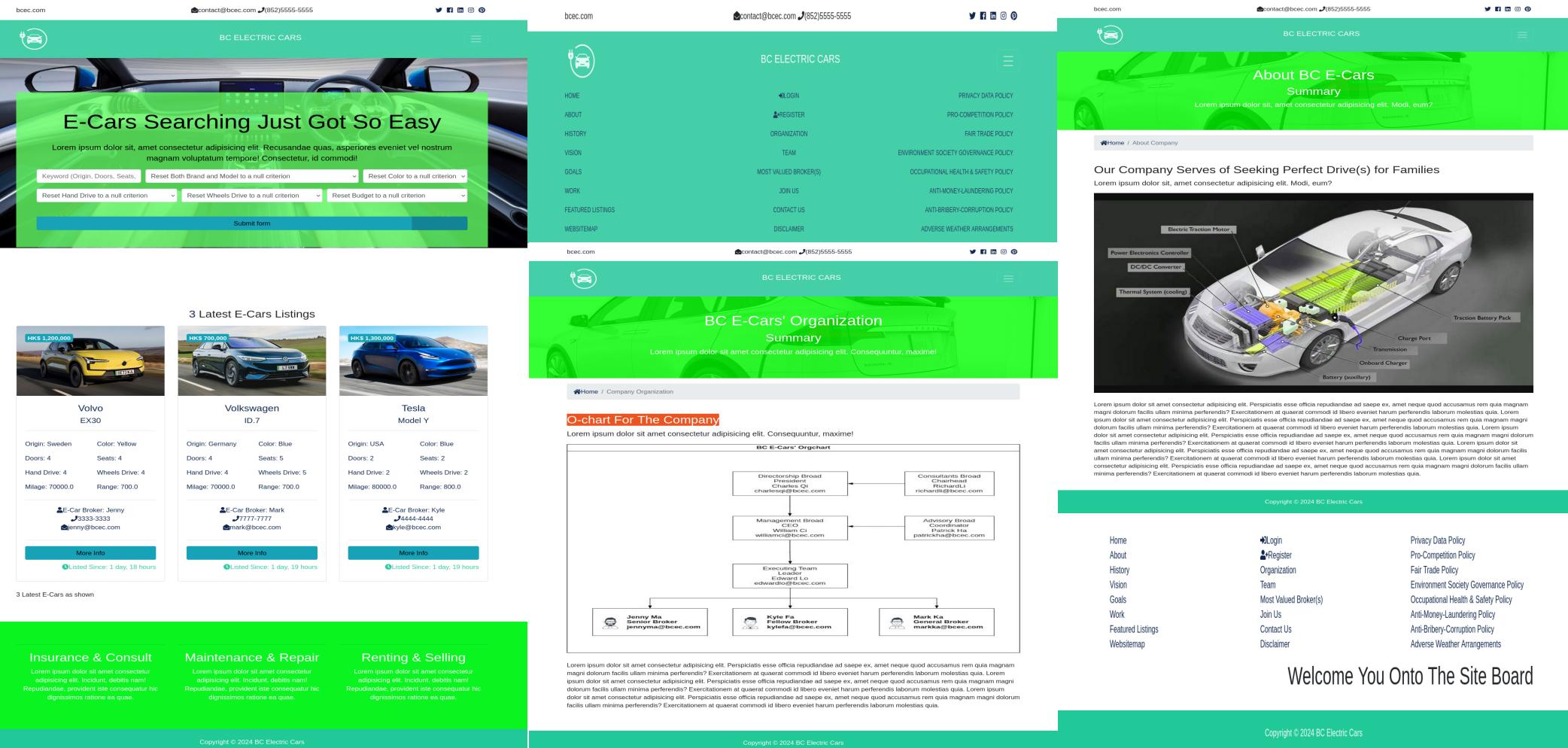


# BC Electric Cars

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.

The header features a top navigation bar with links for Home, About, History, Vision, Goals, Work, Featured Listings, and Webitemap. Below this is a secondary navigation bar with links for Login, Register, Organization, Pro-Competition Policy, Fair Trade Policy, Occupational Health & Safety Policy, Anti-Money Laundering Policy, Anti-Bribery-Corruption Policy, and Adverse Weather Arrangements. The main content area includes sections for E-Cars Searching Just Got So Easy, 3 Latest E-Cars Listings, BC E-Cars' Organization Summary, O-chart For The Company, and various service sections like Insurance & Consult, Maintenance & Repair, Renting & Selling, and Contact Us. A sidebar on the right provides links to the company's privacy and legal policies.

# Enriched NavBar-Left/Center/Right Cols

BC E-Cars | Welcome

localhost:8000

bcec.com

contact@bcec.com (852)5555-5555

bccec.com

BC ELECTRIC CARS

HOME

ABOUT

HISTORY

VISION

GOALS

WORK

FEATURED LISTINGS

WEBSITE MAP

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

E-Cars Searching Just Got So Easy

bcec.com

contact@bcec.com (852)5555-5555

bccec.com

BC ELECTRIC CARS

Login

Username

Password

Login

Copyright © 2024 BC Electric Cars

bccec.com

contact@bcec.com (852)5555-5555

bccec.com

BC ELECTRIC CARS

Register

First Name

Last Name

Username

Email

Password

Confirm Password

Register

Copyright © 2024 BC Electric Cars

# Newly WebSiteMap: logged-in/out differ



[Home](#)  
[About](#)  
[History](#)  
[Vision](#)  
[Goals](#)  
[Work](#)  
[Featured Listings](#)  
[Websitemap](#)

Welcome frankmok (dashboard)  
[Logout](#)  
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-Laundersing Policy  
Anti-Bribery-Corruption Policy  
Adverse Weather Arrangements

Welcome You Onto The Site Board

Copyright © 2024 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-Laundersing Policy  
Anti-Bribery-Corruption Policy  
Adverse Weather Arrangements

Welcome You Onto The Site Board

Copyright © 2024 BC Electric Cars

# Enabled Reset Some Search Fields

The screenshot shows a web browser window for 'BC E-Cars | Welcome' at 'localhost:8000'. The page has a teal header with a car icon, contact info ('contact@bcec.com', '(852)5555-5555'), and social media links. The main content features a blurred image of a car interior and the text 'E-Cars Searching Just Got So Easy'. Below this is a green overlay containing placeholder text and several dropdown menus for search criteria: 'Keyword (Origin, Doors, Seats, ...)', 'Reset Both Brand and Model to a null criterion', 'Reset Color to a null criterio...', 'Reset Hand Drive to a null criterio...', 'Reset Wheels Drive to a null criterio...', 'Reset Budget to a null criterio...', and a 'Submit form' button.

BC E-CARS

E-Cars Searching Just Got So Easy

Reset Both Brand and Model to a null criterion

Reset Hand Drive to a null criterio...

Reset Wheels Drive to a null criterio...

Reset Budget to a null criterio...

Submit form

# UX Design – Realisticity in Search

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

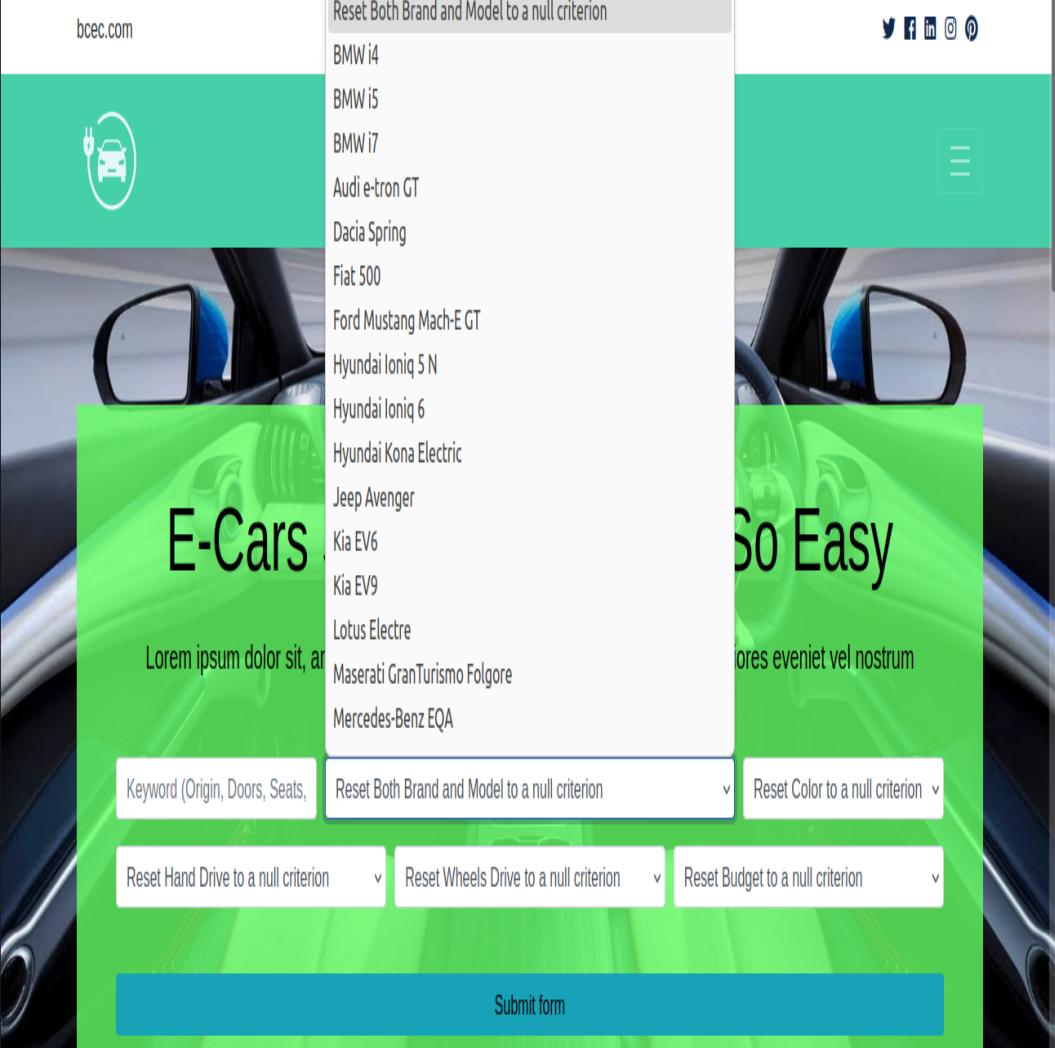
UX can be increased by offer of more realisticity senses to users in the options' choices.

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



```
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 users' unrealistic combinations to drive out possible confusions in selecting options to keep up their rigorous sense.

The screenshot shows a web page for "E-Cars" (bcec.com) featuring a search form. A dropdown menu is open over the "Brand" input field, listing various electric vehicle models and brands. The dropdown title is "Reset Both Brand and Model to a null criterion". The listed items 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 has a green header with social media icons and a sidebar with a car image and text. The main search form includes fields for Keyword, Hand Drive, Wheels Drive, Budget, and a Submit button.

# Combinations of Search Choices / Fields

Other relatively minor search fields / choices in this case such as color and budget can be individualized as of users' preferential freedom allowing for negative combinations for example below a budget not a color available or accessible retaining user's 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)" with a "Reset Both Brand and Model to a null criterion" button; "Reset Hand Drive to a null criterion" and "Reset Wheels Drive to a null criterion" dropdowns; and a "Reset Budget to a null criterion" dropdown which is currently open, displaying a list of price points from HK\$200,000 to HK\$2,200,000 in increments of HK\$200,000. A large blue "Submit form" button is at the bottom.

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

Reset Hand Drive to a null criterion

Reset Wheels Drive to a null criterion

Reset Budget to a null criterion

Submit form

# 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". A large blue button labeled "Submit form" is centered below the dropdowns. To the right, 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 > ...

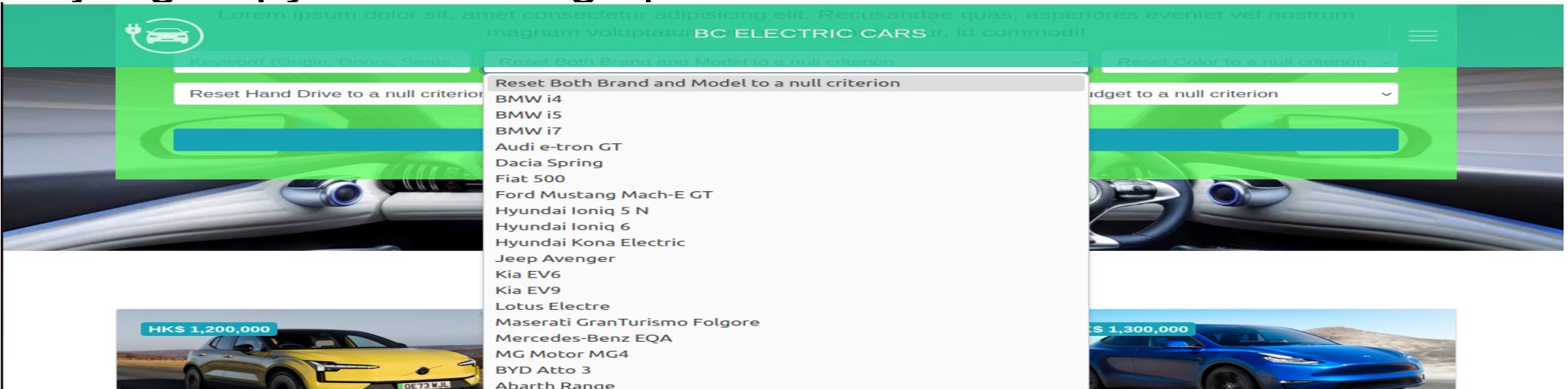
```
3     listings = Listing.objects.all().filter(is_published=True)
10    max_price = max({listing.price for listing in listings})
11    grade = 200000
12    budget_choices = {'Reset Budget to a null criterion': '0'} |_
13    {'HK$'+f'{g:,}': str(g) for g in range(grade, max_price + grade, grade)}
```

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

# django / python coding tips: the websitemap code

templates > pages > sitemap.html

```
46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145<div class="row">
    <div class="col-md-4">
        <ul style="list-style-type:none;">
            <li>
                | <a href="{% url "index" %}>Home</a>
            </li>
            <li>
                | <a href="{% url "about" %}>About</a>
            </li>
            <li>
                | <a href="{% url "history" %}>History</a>
            </li>
            <li>
                | <a href="{% url "vision" %}>Vision</a>
            </li>
            <li>
                | <a href="{% url "goals" %}>Goals</a>
            </li>
            <li>
                | <a href="{% url "work" %}>Work</a>
            </li>
            <li>
                | <a href="{% url "listings" %}>Featured Listings</a>
            </li>
            <li>
                | <a href="{% url "sitemap" %}>Websitemap</a>
            </li>
        </ul>
    </div>
    <div class="col-md-4">
        <ul style="list-style-type:none;">
            &% if user.is_authenticated %
            <li>
                | <a href="{% url "dashboard" %}><i class="fas fa-user"></i>Welcome {{user.username}} (dashboard)</a>
            </li>
            <li>
                | <a href="javascript: {document.getElementById('logout').submit()}">
                    | <i class="fas fa-sign-out-alt"></i>Logout</a>
                <form action="{% url 'logout' %}" method="POST" id="logout">
                    {% csrf_token %}
                    | <input type="hidden" />
                </form>
            </li>
            &% else %
            <li>
                | <a href="{% url "login" %}><i class="fas fa-sign-in-alt"></i>Login</a>
            </li>
            <li>
                | <a href="{% url "register" %}><i class="fas fa-user-plus"></i>Register</a>
            </li>
        &% endif %
    </div>
    <div class="col-md-4">
        <ul style="list-style-type:none;">
            <li>
                | <a href="{% url "organization" %}>Organization</a>
            </li>
            <li>
                | <a href="{% url "team" %}>Team</a>
            </li>
            <li>
                | <a href="{% url "mvb" %}>Most Valued Broker(s)</a>
            </li>
            <li>
                | <a href="{% url "joinus" %}>Join Us</a>
            </li>
            <li>
                | <a href="{% url "contactus" %}>Contact Us</a>
            </li>
            <li>
                | <a href="{% url "disclaimer" %}>Disclaimer</a>
            </li>
        </ul>
    </div>
    <div class="col-md-4">
        <ul style="list-style-type:none;">
            <li>
                | <a href="{% url "pdpolicy" %}>Privacy Data Policy</a>
            </li>
        </ul>
    </div>
</div>
```

# 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.