BÀI 2: Forms

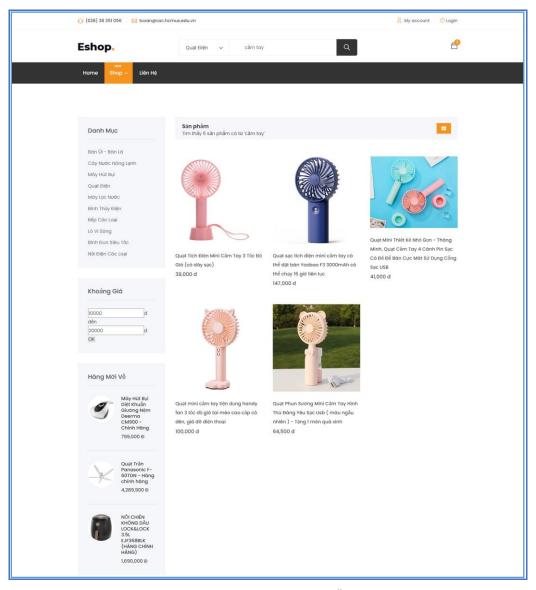


Mục tiêu chính:

- Các kiến thức cơ bản về form
- Tao các form trong ứng dụng web

2.1. Form tìm kiếm sản phẩm

√ Yêu cầu: Tạo form và thực hiện việc tìm kiếm sản phẩm theo loại và theo nội dung người dùng nhập:



Tìm kiếm sản phẩm: Loại "Quạt điện" – Chuỗi tìm kiếm: "Cầm tay"

✓ Hướng dẫn

- Sử dụng project MyStore được cung cấp hoặc tiếp tục sử dụng project MyStore ở bài trên.
- Trong store > views.py > tạo function search_form(): thực hiện công việc đọc, lọc dữ liệu theo yêu cầu nhận được từ form và gửi kết quả ra store/shop.html. Chú ý: Viết code cho cả hai trường hợp là người dùng chọn một SubCategory cụ thể hoặc chon "Tất cả".

```
def search_form(request):
   global subcategory
   global search_str
    product items = 0
   three_newest = models.Product.objects.all().order_by("-
public_day")[:3]
   product list = []
    if request.method == 'GET':
        form = forms.FormSearch(request.GET, models.Product)
        if form.is valid():
            subcategory = form.cleaned data['subcategory id']
            search_str = form.cleaned_data['name']
            if subcategory != 0:
                product list = models.Product.objects.filter(
                    subcategory=subcategory, name__contains=search_st
r).order_by("-public_day")
            else:
                product list = models.Product.objects.filter(
                    name__contains=search_str).order_by("-
public_day")
        product_items = len(product_list)
        page = request.GET.get('page', 1)
        paginator = Paginator(product_list, 9) # so product/trang
       try:
            products = paginator.page(page)
        except PageNotAnInteger:
            products = paginator.page(1)
        except EmptyPage:
            products = paginator.page(paginator.num_pages)
        return render(request, "store/shop.html",
                      {'three newest': three newest,
                       'subcategories': subcategory list,
                       'products': products,
```

```
'pk': subcategory,
    'subcategories': subcategory_list,
    'product_items': product_items,
    'subcategory': subcategory,
    'search_str': search_str
})
_
```

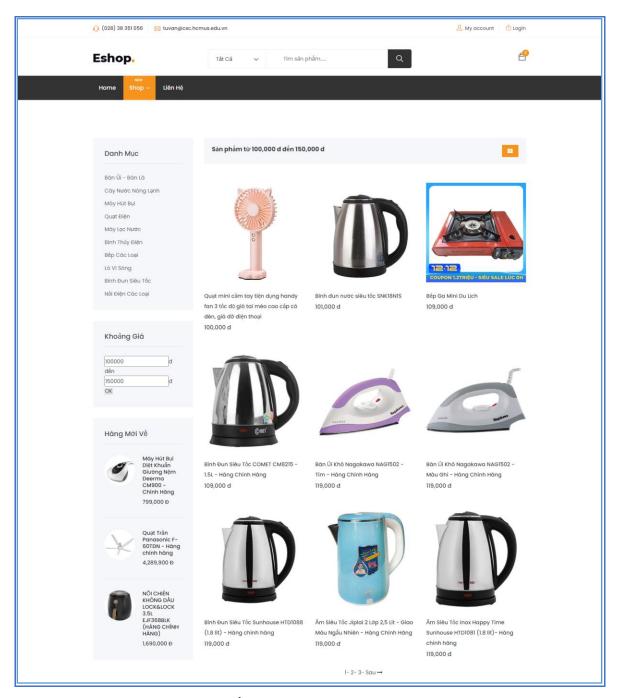
- Khai báo url của function search_form() trong tập tin urls.py của app store.
- Trong template base.html: Thiết kế form "Tìm sản phẩm":

```
<div class="col-lg-8 col-md-7 col-12">
    <div class="search-bar-top">
       <form action="{% url 'store:search.html' %}" method="GET" name="search_form">
           <div class="search-bar">
                <select style="width:auto;" name="subcategory_id">
                   <option selected="selected" value="0">Tất cả</option>
                   {%if subcategories %}
                   {% for category in subcategories%}
                   {% if subcategory == category.id %}
                    <option value="{{category.id}}" selected">{{category.name}}</option>
                   {% else %}
                   <option value="{{category.id}}">{{category.name}}</option>
                    {% endif %}
                    {% endfor %}
                    {% endif %}
                </select>
                <input name="name" placeholder="Tim sån phẩm...." type="search"</pre>
                   value="{{search str}}">
                <button class="btnn"><i class="ti-search"></i></button>
            </div>
        </form>
    </div>
</div>
```

- Bổ sung một số dòng code trong shop.html cần thiết để hiển thị kết quả tìm kiếm
- Chạy thử và kiểm tra kết quả

2.2. Form lọc sản phẩm theo giá

√ Yêu cầu: Thiết kế form lọc sản phẩm theo giá



Các sản phẩm có giá từ 100.000 đ => 150.000 đ

✓ Hướng dẫn

- Tiếp tục sử dụng project MyStore ở bài trên
- Trong store > views.py > tạo function filter_by_prices(): thực hiện công việc đọc, lọc dữ liệu theo giá người dùng nhập từ form và gửi kết quả ra store/shop.html.
- Chú ý: khi viết code cần có đủ các trường hợp: người dùng nhập giá khi trang shop.html đang hiển thị tất cả các sản phẩm thì lọc giá theo tất cả các sản phẩm, khi trang shop đang hiển thị danh sách sản phẩm theo loại thì lọc giá theo các sản phẩm

thuộc loại đó, khi trang shop đang hiển thị danh sách sản phẩm theo kết quả tìm kiếm thì lọc giá theo danh sách các sản phẩm tìm kiếm.

```
def filter_by_prices(request):
   global subcategory
   global search str
   product_items = 0
   product list = models.Product.objects.all().order by("price")
   three newest = models.Product.objects.all().order by("-
public_day")[:3]
   result = ""
    if request.method == 'GET':
        subcategory = request.GET.get('subcategory id 1')
        x = 'yes'
        if request.GET.get('name 1'):
            search str = request.GET.get('name 1')
        else:
            search_str =''
            x = 'no'
        a = float(request.GET.get('price from'))
        b = float(request.GET.get('price to'))
        price from = a
        price to = b
        if(a > b):
            price from = b
            price to = a
        if subcategory != '0':
            result = "SubCategory " + \
                    str(subcategory) + " - " + \
            str(price_from) + " - " + str(price_to)
            if(x!='no'):
                product list = models.Product.objects.filter(
                    name contains=search str, subcategory=subcategor
y).order_by("price")
            else:
                product_list = models.Product.objects.filter(subcateg
ory=subcategory).order_by("price")
         if subcategory == '0':
            result = "SubCategory = 0" + str(subcategory)
            if(x!='no'):
                product_list = models.Product.objects.filter(name__co
ntains=search_str).order_by("price")
        product list = [product for product in product list if product
t.price >= price from and product.price <=price to]</pre>
```

```
product_items = len(product_list)
      result = 'từ ' + '{:20,.0f}'.format(price_from) + ' đ đến ' +
'{:20,.0f}'.format(price_to) + ' d'
      page = request.GET.get('page', 1)
      paginator = Paginator(product_list, 9) # so product/trang
      try:
          products = paginator.page(page)
      except PageNotAnInteger:
          products = paginator.page(1)
      except EmptyPage:
          products = paginator.page(paginator.num_pages)
      return render(request, "store/shop.html",
                     {'three_newest': three_newest,
                      'subcategories': subcategory_list,
                      'products': products,
                      'pk': subcategory,
                      'subcategories': subcategory_list,
                      'product_items': product_items,
                      'subcategory': subcategory,
                      'search_str': search_str,
                      'result': result
                      })
```

- Khai báo url của function filter_by_prices() trong tập tin urls.py của app store.
- Bổ sung một số dòng code trong shop.html cần thiết để hiển thị kết quả tìm kiếm
- Chay thử và kiểm tra kết quả.