

BÀI 2: Forms



Mục tiêu chính:

- Các kiến thức cơ bản về form
- Tạo 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:**

The screenshot shows the Eshop website interface. At the top, there's a header with contact information, a search bar, and navigation links. The main content area displays search results for 'cầm tay' (handheld) fans. On the left, there's a sidebar with filters: 'Danh Mục' (Categories) listing various home appliances, 'Khoảng Giá' (Price Range) with input fields for 10000 and 20000, and 'Hàng Mới Về' (New Arrivals) featuring three products: a vacuum cleaner, a Panasonic fan, and a rice cooker. The main product grid shows five handheld fans with their descriptions and prices: 39,000 đ, 147,000 đ, 41,000 đ, 100,000 đ, and 64,500 đ.

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 chọn "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_str).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="selected">{{category.name}}</option>
                    {% else %}
                    <option value="{{category.id}}">{{category.name}}</option>
                    {% endif %}
                    {% endfor %}
                    {% endif %}
                </select>
                <input name="name" placeholder="Tìm sản phẩm....." type="search"
                    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**

(028) 38 351 056

tuvan@hcmus.edu.vn

My account

Login

Eshop.

Tất cả

Tìm sản phẩm...

Home

Shop

Liên Hệ

Danh Mục

Bàn Ủi - Bàn Lò

Cây Nước Nóng Lạnh

Máy Hút Bụi

Quạt Điện

Máy Lọc Nước

Bình Thủy Điện

Bếp Cốc Loại

Lò Vi Sóng

Bình Đun Siêu Tốc

Nồi Điện Cốc Loại

Sản phẩm từ 100,000 đ đến 150,000 đ

Quạt mini cầm tay tiện dụng handy fan 3 tốc độ gió tai mèo cao cấp có đèn, gió đỡ điện thoại

100,000 đ

Bình đun nước siêu tốc SNK18N1S

101,000 đ

Bếp Ga Mini Du Lịch

109,000 đ

Khoảng Giá

100000 đ

đến

150000 đ

OK

Hàng Mới Về

Máy Hút Bụi Diệt Khuẩn Giường Nệm Deerma CM900 - Chính Hãng

799,000 đ

Quạt Trán Panasonic F-60TDN - Hàng chính hãng

4,289,900 đ

NỒI CHIÊN KHÔNG DẦU LOCK&LOCK 3.5L EJF368BLK (HÀNG CHÍNH HÃNG)

1,690,000 đ

Bình Đun Siêu Tốc COMET CM8215 - 1.5L - Hàng Chính Hãng

109,000 đ

Bàn Ủi Khô Nagakawa NAG1502 - Tím - Hàng Chính Hãng

119,000 đ

Bàn Ủi Khô Nagakawa NAG1502 - Màu Ghi - Hàng Chính Hãng

119,000 đ

Bình Đun Siêu Tốc Sunhouse HTD1088 (1.8 lít) - Hàng chính hãng

119,000 đ

Ấm Siêu Tốc Jiplai 2 Lớp 2.5 Lit - Giao Màu Ngẫu Nhiên - Hàng Chính Hãng

119,000 đ

Ấm Siêu Tốc Inox Happy Time Sunhouse HTD1081 (1.8 lít) - Hàng chính hãng

119,000 đ

1 - 2 - 3 - Sau

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=subcategory
                ).order_by("price")
            else:
                product_list = models.Product.objects.filter(subcategory=subcategory).order_by("price")
            if subcategory == '0':
                result = "SubCategory = 0" + str(subcategory)
            if(x!='no'):
                product_list = models.Product.objects.filter(name__contains=search_str).order_by("price")

        product_list = [product for product in product_list if product.price >= price_from and product.price <=price_to]
```

```

        product_items = len(product_list)

        result = 'từ ' + '{:20,.0f}'.format(price_from) + ' đ đến ' +
        '{:20,.0f}'.format(price_to) + ' đ'

        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
- Chạy thử và kiểm tra kết quả.