



Travaux Pratiques n°4

Framework Django - Les templates

Nous allons travailler dans ce tp sur le même projet 'mysite et la même application 'magasin' créée dans les tp précédents.

Partie 1:

1- créer le répertoire **templates** sous votre application et configurer votre **setting.py** dans la rubrique TEMPLATES.

```
TEMPLATES = [  
    {  
        'BACKEND': 'django.template.backends.django.DjangoTemplates',  
        'DIRS': [os.path.join(BASE_DIR, 'templates')]  
    },  
    {  
        'APP_DIRS': True,  
        'OPTIONS': {  
            'context_processors': [  
                'django.template.context_processors.debug',  
                'django.template.context_processors.request',  
                'django.contrib.auth.context_processors.auth',  
                'django.contrib.messages.context_processors.messages',  
            ],  
        },  
    },  
]
```

2- Dans le répertoire **templates**, créer tout d'abord un dossier **magasin** (le nom de l'application) car nous aurons à créer dans toutes les applications un dossier **templates** alors pour différencier ces dossier nous ajoutons un sous repertoire qui porte le nom de l'application dans lequel seront places tous les templates. Puis créer un template **mesProduits.html** sous **templates/magasin** permettant de visualiser la liste des produits.

```
<h2>Mes articles..... </h2>  
{% for prod in products %}  
    <h2>{{prod.libelle}}, {{prod.prix}}</h2>  
{% endfor %}
```

3- Configurer **views.py** avec le nouveau template :

```
from django.shortcuts import render  
from django.template import loader  
from .models import Produit  
def index(request):  
    products= Produit.objects.all()  
    context={'products':products}
```

```
return render( request,'magasin/ mesProduits.html ',context ))
```

4- Configurer **urls.py** :

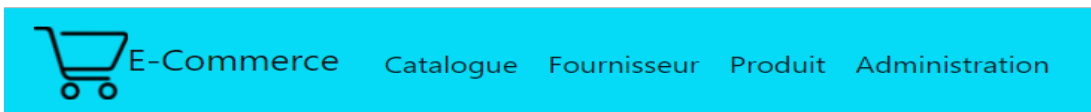
```
from django.urls import path
from . import views
urlpatterns = [
    path("", views.index, name='index'),
]
```

Partie 2 :

1- créer les répertoires **static/css** sous l'application **magasin**. Dans le dossier **css** vous mettez le fichier **base.css** suivant :

```
#conteneur
{
    display: flex;
    justify-content: center;
    align-items: center;
}
```

2- Nous voulons créer un template **base.html** sous **templates/magasin** contenant la barre de menu de l'application et tous les composants qui se répètent dans les pages du site;



pour cela, adopter la squelette suivante :

```
{% load static %}
<!DOCTYPE html>
<html>
<head>
    <meta charset='utf-8'>
    <meta http-equiv='X-UA-Compatible' content='IE=edge'>
    <title>
        {% block title %} {% endblock %}
    </title>
    <link rel='stylesheet' type='text/css' media='screen'
        href="{% static 'css/base.css'%}">

    {% block styles %} {% endblock %}
</head>
<body>
    {% block content %} {% endblock %}
</body>
</html>
```

Puis :

- Dans la partie **<body>**, insérer un menu **nvbar** en utilisant **Bootstrap**;

👉 Le lien **href** entre l'option du menu "Navbar" et le traitement correspondant se fait à travers l'option "name=index" indiquée dans le fichier **views.py**.

```
<li class="nav-item active">
<a class="nav-link" href="{% url 'index' %}">Catalogue <span class="sr-only">(current)</span></a>
</li>
```

NB : pour l'affichage des images, assurez vous d'avoir ajouter dans **settings.py** :

```
STATIC_URL = '/static/'
STATICFILES_DIRS = [BASE_DIR / "static", '/var/www/static/', ]
```

3- créer **mesProduits.html**, un template dérivé de **base.html** et qui affiche la liste de tous les articles d'une manière simplifiée.

```
{% extends 'magasin/base.html' %}
{% block title %} Catalogue de produits {% endblock title %}
{% block content %}
<!-- body -->
<div class="container-fluid mt-3">
  <div class="row">
    <!-- all products -->
    <div id='products' class="col-lg-9 mx-auto">
      <div class="row mx-auto">
< ! -- Products est une clé du dictionnaire context rendu par la vue 'index' -->
        {% for product in products %}
          <div id={{product.id}} style="width: 18rem;">
            
            <div >
              <p >{{product.libelle}}</p>
              <p ><b>{{product.description}}</b></p>

            </div>

          </div>
        {% endfor %}
      </div>
    </div>
  </div>
</div>

{% endblock %}
```

A ce stade, vous devez obtenir une interface similaire à celle ci :



couette

200x200 double face



tomate

tomate boîte 800gr



table

table ronde