

Travaux Pratiques n°12

Framework Django - Les Formulaires

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

Partie 1:

Pour le template **mesProduits.html** du TP précédent, nous allons créer un formulaire pour afficher tous les produits avec une interface plus ergonomique.

1- Dans un fichier **forms.py**, préparer votre formulaire **ProduitForm** permettant de récupérer toutes les données de la table **Produit**.

```
from django.forms import ModelForm
from .models import Produit
class ProduitForm(ModelForm):
    class Meta :
        model = Produit
        fields = "__all__" #pour tous les champs de la table
        #fields=['libelle','description'] #pour quelques champs
```

2- Modifier **views.py** de telle sorte que la méthode **index()** permet de créer une instance du formulaire et de l'envoyer à l'URL du template **majProduits.html** pour son affichage.

```
from .models import Produit
from .formulaire import ProduitForm
from django.shortcuts import redirect

def index(request):
    if request.method == "POST" :
        form = ProduitForm(request.POST)
        if form.is_valid():
            form.save()
            return HttpResponseRedirect('/magasin')
    else :
        form = ProduitForm() #créer formulaire vide
        return render(request,'magasin/majProduits.html',{'form':form})
```

3- créer un template **majProduits.html** dans le repertoire **templates/magasin** qui hérite de **base.html**. **majProduits.html** doit définir le contenu de la balise générique **{% block body %}** du template **base.html** créé pour afficher les informations envoyées par **views.py**.

```
{% extends 'magasin/base.html' %}
{% block body %}
    <div class="col-md-12 text-center">
        <h2>Manip Produit ..... </h2>
    </div>
    <form method='POST' class="post-form">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit" class="btn btn-default">Enregistrer</button>
    </form>
{% endblock %}
```

Testez et assurez vous de l'affichage du formulaire.

Ajoutez un enregistrement et vérifiez son insertion dans la base.

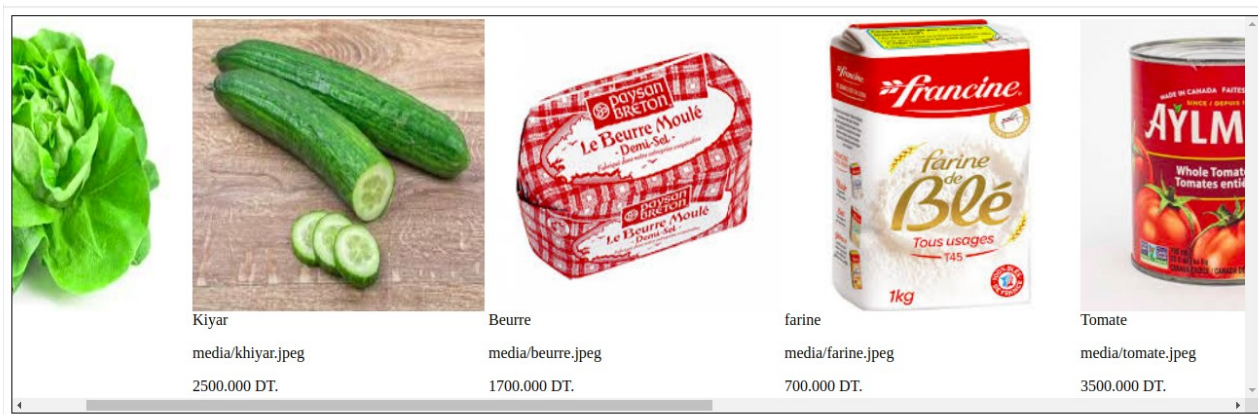
Partie 2:

1- Modifier la méthode **index()** de **views.py** comme suit :

```
list=Produit.objects.all()
return render(request,'vitrine.html',{'list':list})
```

permettant de récupérer tous les produits de la base.

2- Pour améliorer l'affichage de la liste des produits, créer un template **vitrine.html** qui hérite de **base.html** et permettant de présenter tous les produits du magasin **avec ascenseur de défilement**.



```
{% extends 'base.html' %}
{% load static %}
{% block body %}
    <form method='POST' class="post-form">
        {% csrf_token %}
        <div class="container">
            <div style="overflow:scroll; border:#000000 1px solid;">
```

```

<table>
  <tr>
    {% for article in list %}
      <td>
        <img src='/media/{{ article.img }}'

          class="rounded float-start"
          height="200px"
          width="200px"

          alt="">
        <figcaption class="figure-caption">
          {{article.libelle}}
        </figcaption>
        <p> {{article.img}} </p>
        <p> {{article.prix}} DT. </p>
      </td>
    {% endfor %}
  </tr>
</table>
</div>
</form>

</div>
{% endblock %}

```

Testez l’affichage.

3- Pensez à créer le template et le formulaire de la commande.

Partie 3:

Remarquez jusqu’à maintenant, dans le projet ‘myproject’, nous avons créé une seule application ‘magasin’. Dans le cas pratique, un projet peut avoir plusieurs applications.

Pour cela, nous voulons créer une page d’accueil au projet qui permettra de naviguer entre les applications.

1- Créer un fichier **accueil.html** sous un répertoire **templates** sous le répertoire racine du projet:

```

<!DOCTYPE html">
<html lang="fr">
<head>
<meta charset="utf-8" />
<title>Création d'un design étape par étape - Etape n°1 : Le code xhtml</title>
</head>
<body>
<div id="conteneur">
<h1 id="header"><a href="#" title="Colored Design - Accueil"><span>Colored
Design</span></a></h1>

```

```

<nav>
<ul id="menu">
<li><a href="/mag/">Application n°1 : Magasin</a></li>
<li><a href="#">Application n°2</a></li>
<li><a href="#">Application n°3</a></li>
</ul>
</nav>
<div id="contenu">
<h2>Un titre</h2>
<p>Un paragraphe</p>
</div>
<p id="footer">Réalisation des codes HTML/CSS, du tutoriel et du design par ElMoustiko</p>
</div>
</body>
</html>

```

2- Créer le fichier **views.py** sous le projet :

```

from django.shortcuts import render
from django.template import loader

def index(request):
    return render(request,'accueil.html' )

```

3- dans **urls.py** du projet, ajouter la ligne suivante :

```

urlpatterns = [
    path('admin/', admin.site.urls),
    path('magasin/', include('magasin.urls')),

    path("", views.index)
]+ static(settings.MEDIA_URL, document_root= settings.MEDIA_ROOT)

```

4- Le fichier **setting.py** devient :

```

TEMPLATES = [
{
    'BACKEND': 'django.template.backends.django.DjangoTemplates',
    'DIRS': ['magasin/templates/magasin',
             'mysite2022/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',
            'django.template.context_processors.media',
        ],
    },
},

```

},
1