Université des Sciences et de la Technologie Houari Boumediene
Faculté d'Informatique
Département des systèmes informatiques

Module: Systèmes d'information 2 **L3.ISIL.A**
Chargé de TP : R.BOUDOUR **2023/2024**

TP N°5 : Utilisation des formulaires dans Django

Partie 1: La recherche dans la BDD

Objectif : Une fois que vos modèles sont créés et qu'un ensemble de leurs instances est présentée aux utilisateurs (clients) à l'aide des Templates. Il est temps d'écrire le code qui permet aux clients de réagir et interagir avec les informations affichées (rechercher une information, insérer une donnée, etc.). Vous devrez donc implémenter des formulaires comprenant des champs de saisie pour les utilisateurs de votre application.

Partie 1: Création d'un formulaire à la main (HTML forms) :

Nous allons nous intéresser dans la première partie à la création manuelle des formulaires. Vous allez utiliser des balises HTML pour insérer un formulaire de rechercher qui permettra aux clients de rechercher l'existence d'un produit dans votre BDD. Les fichiers que vous allez modifier sont : ***urls.py*** - ***views.py*** et templates / ***search.html***

Travail demandé :

Afin d'afficher tous les produits correspondant à une recherche demandée par un client :

1- Modifiez le fichier: ***search.html*** *(templates >> search.html)* pour insérer un formulaire permettant la recherche.

```
<form action="" method="GET">
       <label for="">Entrez le nom du produit</label>
       <input type="text" name="search" id="">
       <button type="submit"> démarrer la recherche</button>
 </form>
```

2- Modifiez le fichier ***urls.py*** (de l'application) pour définir : l'***url*** sur laquelle vous répondrez à la demande du client (afficher le résultat de la recherche) & la ***fonction*** (vue) qui répondra à la demande (recherche de produits).

```
urlpatterns = [  . . . ,
    path('search_product/',views.rechercher_produits, name="listing"),
  ]
```

- Il est clair que la fonction ***rechercher_produits()*** n'a pas encore été définie, pour la définir il faut aller sur ***views.py.***

R.BOUDOUR| rboudour@usthb.dz

3- Modifiez le fichier ***views.py*** pour permettre d'interagir avec la BDD (***models.py)*** et de répondre à la requête. La réponse sera ensuite envoyée à un Template (***search.html***) pour être affichée aux utilisateurs.

```
def rechercher_produits(request):
    if request.method == "GET":
        query=request.GET.get('search')
        if query:
            produits=Product.objects.filter(prd_name__contains=query)
            return render(request,'search.html', {'products': produits })
        return render(request,'search.html')
```

4- Modifiez le fichier ***search.html*** pour récupérer les données envoyées par la fonction *rechercher_produits()* et insérez-les dans votre page html. Apres avoir les insérer, ces données seront accessibles à tous les utilisateurs à partir de l'url : http://127.0.0.1:8000/search_product/

```
<h4> resultat de recherche : </h4>
<ul>
{% for product in products %}
    <li> Nom du produit: {{ product.prd_name }},
         son prix: {{ product.prd_price }} Da. </li>

  {% endfor %}
</ul>
```

Bonus:

Modifiez votre code de recherche pour couvrir le scénario suivant :

« Si le résultat correspondant à une recherche demandée est vide (pas de produit correspondant), élargissez-la en explorant les ingrédients du produit pour afficher ceux contenant la chaîne recherchée en tant qu'ingrédient ».

L'affichage sera de la forme :

Recherche niveau 1 – Aucun produit n'a été trouvé !

Résultat de recherche niveau 2 :

- **Produit 1**
- **. . .**

R.BOUDOUR| rboudour@usthb.dz