

2013

Nuit de L'info
Défi Constrictor Python, Django
& Co

Equipe e-Nova
Robotics

I)Introduction

II)Concept

III)Réalisation

I)Introduction

Nous avons identifié le défi qui consistait à utiliser le python comme un langage coté serveur avec le Framework Web Django. C'est une solution alternative au php et java qui sont plus compliqués.

Le but de ce défi est de tirer le meilleur profit de cette nouvelle technologie qui monte par sa simplification des fonctions.

Ce défi sera en collaboration avec le défi de faire un site Full Ajax (coté client) qui a pour but de gérer les animations et effets lors de l'action du client sur le site.

II)Concept :Fonctionnalités

Le concept du site est de sortir le client du format des sites commerciaux qui sont de noyer le consommateur d'information. Le but de ce site est de rendre une interface simple à l'utilisateur et au lieu d'afficher les textes bruts. Nous avons eu l'idée d'utiliser deux façons pour afficher les groupes d'articles et articles.

Du point vu du client, il aura deux moyens pour accéder à un article :

-Soit de cliquer sur un des boutons pour visualiser les articles

-Soit de cliquer sur une image sur une étagère afin d'y accéder au groupe d'article puis à l'article.

Malheureusement, nous n'avons pas eu le temps d'intégrer cette fonctionnalité intégralement car cela prend trop de temps pour mettre en place cette application. Nous ne pouvons qu'aller dans un sous-groupe et non dans l'article.

III)Réalisation

Côté serveur, nous utilisons Python avec le Framework Django pour gérer la base de donnée en SQLite. Django réceptionne les requêtes HTTP et renvoi une réponse HTTP en utilisant AJAX.

La base de donnée contient des catégories pouvant être aussi une sous catégorie, chaque catégorie utilise le principe de la clé étrangère pour désigner la catégorie de niveau supérieur. La base de données est aussi prévu pour contenir les articles de manière très sommaire, nous n'avons malheureusement pas eu le temps d'implémenter cette fonctionnalité au sein de notre site.

Nous avons déployer un serveur de test sur un de nos PC pour tester notre site, nous utilisons un serveur Apache dans lequel Django est lancé pour interpréter les requêtes HTTP. Les fichiers statiques tels que les CSS, les fichiers Javascript et les images sont envoyés aux clients par le serveur web Apache. Nous avons aussi configurer le routage des URLs au sein de Django pour que la bonne application Web soit lancée, i.e. la boutique.

Les échanges entre le client et le serveur se font au format JSON, le client demande les sous catégories de la catégorie choisie par l'utilisateur en envoyant une requête GET au serveur, le serveur répond en envoyant les données au format JSON après leur extraction de la base de données en utilisant le module ORM de Django.

Lien du site : fscorpio.pythonanywhere.com