ETNA
3 février 2017

MARMITON

Kherfi_a Lamech_h

Table des matières

Preparation du projet

Rajouter le site dans le host :

sudo vi /etc/hosts ==> dedans on rajoute : 127.0.0.1 marmiton.local

Crée un virtualhost :

chemin et command pour le crée ==> sudo vi /private/etc/apache2/other/ marmiton.conf OU rajouter directement dans le : /etc/apache2/extra/httpd-vhosts.conf mais avant il faut faire un backUp du fichier de config : sudo cp httpd-vhosts.conf httpd-vhosts.conf.bak

le contenu du fichier :

```
<VirtualHost *:80>
  ServerName     marmiton.local
  DocumentRoot /Users/zkherfi/Sites/EtnaSchool/marmiton
  DirectoryIndex index.php
  ErrorLog "/private/var/log/apache2/marmiton-error_log"
  CustomLog "/private/var/log/apache2/marmiton-access_log" common
  <Directory /Users/zkherfi/Sites/EtnaSchool/marmiton>
    AllowOverride All
    Allow from All
  </Directory>
  <Directory>
    AllowOverride All
    Allow from All
  </Directory>
</VirtualHost>
```

-

Après faire ça il faut Restart Apache avec la command : sudo apachectl restart

Architecture MVC

: Controller :

: Core :

: Form :

: Materielize :

: Models :

: Tools :

: Views :

: Scr :

Description des classes principals

Classe mère :

On commence par vous présentez le Core qui contient les Class qui vont être hériter par les Controllers et les Models; et son ***réutilisable pour d'autre projet.*** Dedans on trouve les Classes suivante :

AbstractController : Cette classe gère la mise en relation des Controller avec les Views et ainsi que passer des variables en tableau dans les Views.

AbstractForm : Celle-ci 'set' et valide l'entrée des données du formulaire.

Request : enfin, 'Request' gère les appel url (dispatcher). De ce fait, notre URL est composé d'un « controller/action ».

Controllers :

Le controllers relis les les views et les models.

Models :

Chaque action contient ca requête par rapport a la fonctionnalité demander par le l'action du controller

Tools :

regroupé toute les classes supplémentaire si besoin exemple :

StringTools : Conversion de string en camelCase ainsi le contraire.

SendMail : Envois mail de confirmation.

Form :

Contient les classes qui valide les formulaire (formulaire == classe)

Views :

Dans chaque dossier contient une vie (nameDire == nameFile)

Src :

Contient toute les extensions Css, Js, Img que l'on a besoin pour ce projet.

L'organisation des requête

Pour passer les paramètre du formulaire au controller c'est avec l'action=" l'action du controller"

puis renvoyer au model avec une méthode dedans le $_POST en paramètre

Le model reçois les datas pour faire les requête demander puis il return le result au controller

Add recette :

Etape1 : Ajouter le user en premier pour récupéré son 'id' pour que l'on puisse faire l'ajout de la recette.

Etape2 : Avec le id_user, récupéré dans la premier étape, on fait l'ajout de la recette et on récupère son 'id' pour qu'on puissent ajouter les quantité dans la table de jointure "recette_has_ingredients".

Etape3 : on ajoute tout les ingrédient dans la base de données.

Etape4 : après, avec tout les 'id' besoin pour la table jointure, on ajoute les quantités.

Etape5 : pour clôturer cette requête, on envoie un mail de confirmation a l'utilisateur qui à cette recette.