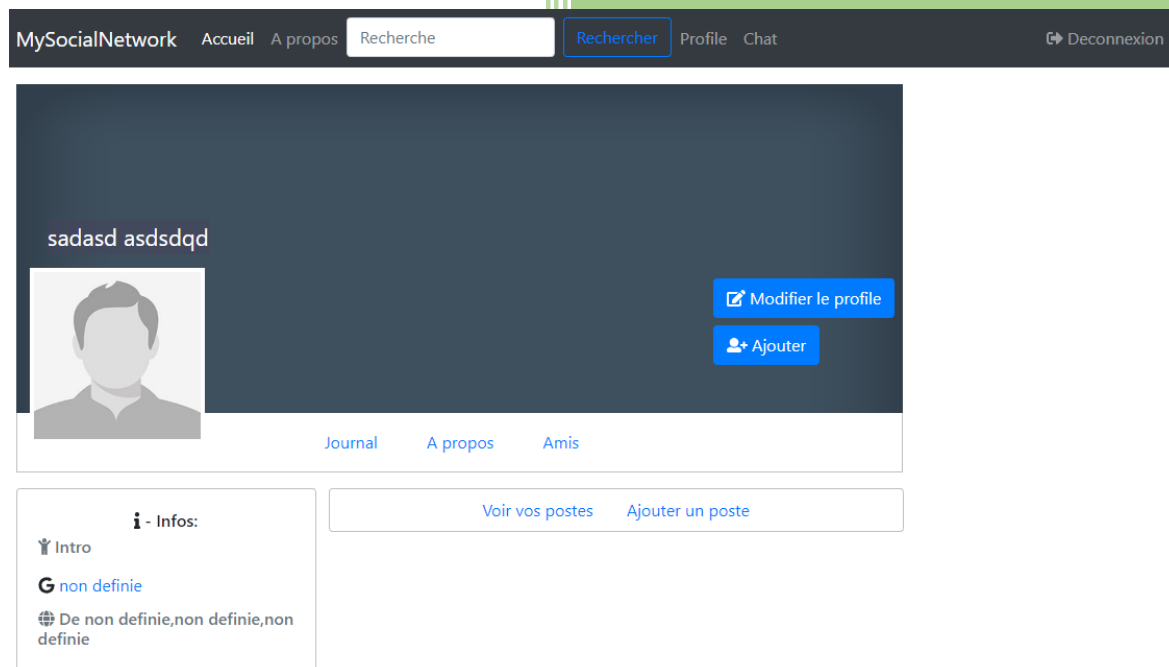


2018

Réseau Social php mysql xml,...



03/03/2018

Nicolas Varlet & Duhamel Antoine

Table des matières

I – Installation de notre site Mysocial Network :	2
A – Prérequis :	2
B – Procédure :	2
1 – Création de la base de donnée :	2
2 – Installation du site :	2
II – Struture MVC :	3
A – Définition :	3
B – Le Modèle :	3
C – Le Contrôleur :	3
D – La vue :	3
E - explications :	4
D - Le router :	4
III – Explication de la base de données :	5

I – Installation de notre site Mysocial Network :

A – Prérequis :

- Une Base de données mysql
- Un serveur web wamp ou lamp CONFIGURE !

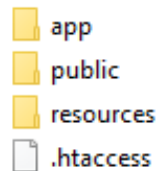
B – Procédure :

1 – Création de la base de donnée :

- o Pour créer la base de donnée il suffit juste d'importer le fichier qui se trouve dans resources nommé mysocialnetwork.sql

2 – Installation du site :

- o Dans le dossier d'installation, la racine du site, vous devriez avoir 3 dossier et un fichier .htaccess comme ci-joint :



- o Une fois cela fait éditer le htaccess qui se trouve dans le dossier public puis éditer la ligne RewriteBase /MySocialNetwork/public et changer comme ceci :
`RewriteBase /ICI_LE_NOM_DU_FICHER_RACINE/public`
- o Maintenant au lieu d'aller dans public allez dans app/config puis éditer config.php et maintenant entrer les informations liées à la base de données précédemment créée ainsi que la ligne suivante :
`define('URLROOT', 'http://localhost/mysocialnetwork');` et changer le liens par votre 'http://ipserver/nom_du_dossier_root_contenant_le_site'
- o Normalement vous pouvez accéder au site
 - En cas de problème n'hésitez pas à me contacter
nicolas.varlet@etud.u-picardie.fr

II – Struture MVC :

A – Définition :

- MVC : Modèle – Vue - Contrôleur
 - o Modèle : gestion des données et des fonctions pour y accéder spécifique à l'application
 - Couche de persistance des données
 - o Vue : affichage des données, elle interagit avec le modèle
 - Interfaces avec l'utilisateur
 - o Contrôleur : synchronisation entre la vue et les données
 - Application, traitements, flux de données
 - S'occupent de la réécriture des URL
 - Déterminé quels traitements doivent être réalisés

B – Le Modèle :

- Le modèle c'est la partie logique de notre application en effet, il comporte les algorithmiques de l'application
- C'est lui qui interagit avec la base de données (SELECT, INSERT, CREATE, READ, UPDATE, DELETE)
- Il communique avec le controller
- Il peut parfois mettre à jour une view

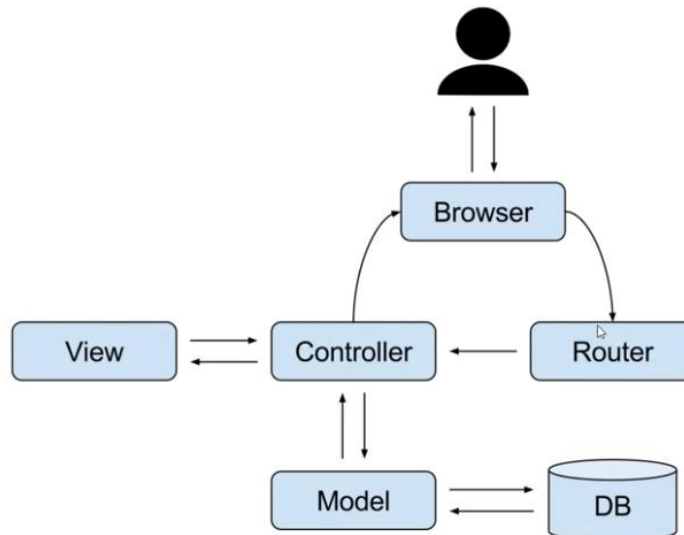
C – Le Contrôleur :

- C'est lui qui reçoit les INPUTS depuis l'URL, des formulaires des views ou autres...
- Il traite les demandes (GET, POST, ...)
- C'est lui qui va chercher les données depuis le Model
- Et c'est aussi lui qui transmet finalement les données aux views

D – La vue :

- Interface graphique où l'utilisateur voit tout ce que le serveur lui renvoie
- Se comporte principalement d'HTML et de CSS
- Lui communique avec le controller
- Il peut recevoir dynamiquement des valeurs grâce au controller

E - explications :



L'utilisateur demande à l'aide de son navigateur une requête au server pour cela :

1 le navigateur va passer par le router/bootstrap a une mission cruciale. En effet

Elle doit rediriger les requêtes reçues vers le bon controller et la bonne method.

2 celui-ci appelle donc le controller, il va dans un premier temps exécuter la method qui lui est proposer avec les attribue qui lui sont associer lors du routage puis il va demander au models des commandes ou appel de fonction ou method.

3 le model lui reçoit l'ordre du controller et donc vas puiser dans la base de donner si nécessaire et sinon vas exécuter l'algorithme demander et le renvoyer au controller

4 le controller reçoit les données du model que l'on stock dans notre mvc dans un tableau associatif nomme \$data Une fois le traitement terminer le controller renvoi au navigateur une view avec le \$data associer.

D - Le router :

- On prend une URL de ce type <http://localhost/projet/Controller/Method/params>
 - Ici Controller renvoie au controller souhaiter
 - La method qui appartient au controller
 - Ou on lui associer en paramètre les valeurs du tableau params
- Comment sa marche ?
 - <http://localhost/projet/Controller/Method/param1/param2>

- L'URL est entièrement découpé et stocké sous forme de tableaux
- `Array([0] => Controller [1] => Method [2] => param1 [3] => param3)`

III – Explication de la base de données :

Pour que notre site soit fonctionnel, il faut d'abord que notre base de données corresponde à celui-ci et que celle-ci soit fonctionnelle.

Notre base de données comporte 7 tables, dans lesquelles on gère les informations de notre réseau social.

Il y a la table « users », où est stockée les informations générales d'un profil.

Ainsi dans cette table on stock l'e-mail, le mot de passe de l'utilisateur (en crypté), date de naissance et d'autres informations nécessaires à l'inscription (question de sécurité, numéro de téléphone, adresse..).

La table « images », est la table qui permet de stocké les images, avec un id image unique, ainsi que l'id de l'utilisateur, son usage, son nom et d'autres attributs, stockée à l'aide d'un blob,

La table « posts », où on gère les posts mis par un utilisateur, avec comme attribut un titre, un texte, et aussi les personnes qui peuvent voir ce post, ainsi qu'une image si on le souhaite.

La table « postcomments » où les informations stockées sont celles des commentaires d'un post (référencer par une id utilisateur et une id post).

De même pour la table « likes » qui réfère grâce à l'id d'un post, l'id de l'utilisateur qui aime ce post.

Il y a la table « messages » qui, pour chaque message est référée par une id, avec comme autres attributs l'id de l'utilisateur destinataire du message ainsi que le message et la date d'envoi de celui-ci.

Ainsi pour chaque réseau social, il faut entretenir une relation avec les autres utilisateurs.

Ici, la table « friendships » gère les relations des utilisateurs, avec un id pour chaque relation, ainsi que les deux id d'utilisateurs et le statut de chaque relation, et la date de début de cette relation.

Grâce à toute ces tables, notre base de donnée est fonctionnelle et permet le bon fonctionnement de notre réseau social.

