

Création d'un système de news (actualités) en PHP

I - Cahier des charges

Les besoins :

- un endroit pour stocker les news
- une page d'administration ajout et suppression de news
- une page pour afficher les news

Les moyens :

- le langage PHP : langage de programmation script et pouvant être orienté objet ;
- base de données MySQL : base de donnée open source.

Niveau : débutant

Pré requis : connaissance de base de PHP, SQL

II - Réalisation

a) Etape 1 : réalisation des tables permettant le stockage des données

Evaluation des besoins et perspectives d'évolution.

Le premier besoin est de créer un endroit permettant le stockage des news, celui-ci doit permettre de les triées.

La table News devra contenir :

- un id (identifiant unique pour chaque news, celui-ci permettant d'identifier chaque news et de pouvoir les triées par la suite) ;
- la date de création, permettant de situer dans le temps les événements ;
- un titre, permettant de résumer l'évènement en quelques mots ;
- la news, le texte contenant l'évènement.

Eventuel rajout :

- la date de modification, permettant de voir quand elle a été éditée ;
- l'auteur, car la table news présentée ci-dessus n'indique pas si la news est de vous ou non,

utile pour un site avec plusieurs rédacteurs ;

- la source, un évènement peut être relaté à partir d'autres articles, il faudra alors les citer.

Création de la table News en SQL

```
CREATE TABLE News (  
  Id_news INT (11) NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  Date_creation INT (10) NOT NULL,  
  Titre VARCHAR(50) NOT NULL,  
  News TEXT NOT NULL  
);
```

Explication

Id_news sert à identifier uniquement chaque news, on met donc ce champ en clé primaire (PRIMARY KEY), ensuite peu importe l'id choisi, on peut donc choisir un id numérique qui s'incrémente automatiquement.

Date_creation stockera la date de création de la news, celle-ci vous verrez plus tard sera stocké sous forme de timestamp UNIX d'où la forme numérique.

Titre sera une chaîne de caractères variables, qui fait l'équivalent de 50 caractères (rarement plus).

News est un type text, car on ne peut pas limiter cette chaîne de caractères à 255 (taille maximale d'un varchar).

Une fois notre table crée il faut pouvoir la remplir, c'est pourquoi nous allons passer à l'étape de la création des enregistrements

b) Etape 2 : réalisation des pages ajout/suppression des news

Tout d'abord, il faut faire une connexion à MySQL (base de données sélectionnée pour l'exemple).

Cette page servira à chaque fois que l'on fera appel à la base de données.

connexion.php :

```
<?php  
$host = 'localhost'; // Adresse de connexion à la base de données ( la  
plupart du temps localhost  
$user = 'root'; // Identifiant de connexion  
$password = ''; // Mot de passe de connexion  
$bdd = 'bdd_news'; // Nom de la base de données à laquelle on va se  
connectée  
  
// Connexion puis sélection de la base de données  
$dblink = mysql_connect($host, $user, $password) ;  
mysql_select_db($bdd, $dblink);  
?>
```

Passons maintenant à la création de l'enregistrement.

Pour cela on va dans un premier temps demander au rédacteur de remplir un formulaire pour la news,

et dans un second temps, enregistrer les données dans la table.

ajout_news.php

```
<?php
include_once('connexion.php'); // Inclusion de la connexion à MySQL

/* Utilisation d'une variable servant n'utiliser qu'une page pour
l'insertion des news
(Elle sert à identifier dans quel état on devra faire l'insertion ou
l'affichage du formulaire) */
$ok = NULL; // Initialisation de la variable ok à la valeur nulle
$ok = $_GET['ok'];

// Si cette variable n'est pas vide ou si la variable existe on peut
insérer l'enregistrement
if(!empty($ok) && isset($ok))
{
    // Récupération des variables
    $titre = $_POST['titre']; // Récupération du contenu d'un formulaire par
la variable $_POST[];
    $news = mysql_real_escape_string($_POST['news'], $dblink);
    // La fonction mysql_real_escape_string() Protège les caractères spéciaux
d'une commande SQL

    // Récupération de la date
    $date_creation = time();

    // Insertion dans la table, on utilise une requete SQL
    $sql = "INSERT INTO VALUES ('', '$titre', '$date_creation', '$news')";
    $query = mysql_query($sql, $dblink); // Envoi de la requête à la base de
données

    unset($ok); // On supprime la variable
}
else
{
    // Sinon on affiche le formulaire
    /* Explication brève du formulaire : déclaration d'un formulaire avec la
balise form,
on renvoi la variable ok avec une valeur quelconque, ici 1 */
    ?>
    <form method="post" action="ajout_news.php?ok=1" name="form_news">
    Titre : <input name="titre" type="text" value="" />
    <br />News : <textarea name="news" rows="30" cols="85"></textarea>
    <br /><input type="submit" value="Ajouter la news" />
    </form>
    <?php
    }
    ?>
```

Nous pouvons créer désormais la page permettant de supprimer ces news (erreur de frappe, news erronée, etc...).

Ici, dans un premier état on va afficher la liste des titres de la news, puis dans un second état la suppression de la news sélectionnée.

suppr_news.php

```
<?php
include_once('connexion.php'); // Inclusion de la connexion à MySQL
```

```

/* Même principe que plus haut, on utilise la variable ok
pour différencier l'état d'affichage et l'état de suppression */
$ok = NULL; // Initialisation de la variable ok à la valeur nulle
$ok = $_GET['ok'];

// Si cette variable n'est pas vide ou si la variable existe on peut
insérer l'enregistrement
if(!empty($ok) && isset($ok))
{
// Récupération de l'id de la news
$id_news = $_GET['id_news'];

// Insertion dans la table, on utilise une requete SQL
$sql = "DELETE FROM news WHERE id_news = ".$id_news."";
$query = mysql_query($sql, $dblink); // Envoi de la requête à la base de
données

unset($ok); // On supprime la variable
}
else
{
    $sql = 'SELECT id_news,titre FROM news';

    $query = mysql_query($sql, $dblink); // Envoi de la requête à la base de
données

    while($fetch=mysql_fetch_array($query))
    {
        // On place en variables le résultat de chaque case du tableau
        $titre = $fetch['titre'];
        $id_news = $fetch['id_news'];

        // Affichage
        echo '<a href="suppr_news.php?id_news='.$id_news.'">'.$titre.'</a>';
    }
}
?>

```

c) Etape 3 : réalisation de la page d'affichage

Maintenant qu'on a crée nos news, il faut faire l'affichage de celles-ci, pour cela on va envoyer une requête SQL, et en récupérer le résultat.

affichage_news.php :

```

<?php
include_once('connexion.php'); // Inclusion de la connexion à MySQL

// Requête SQL
$sql = 'SELECT date_creation,titre,news FROM news ORDER BY date_creation
DESC';
// On sélectionne les champs id_news et titre, qui sont les plus utiles

$query = mysql_query($sql, $dblink); // Envoi de la requête à la base de
données

/* Etant donnée que l'on aura plus d'un enregistrement dans la table,
il faut effectuer une boucle pour afficher tout les résultats.
Tant que l'on aura des enregistrements, on boucle,

```

```

on place dans la variable $fetch le résultat de mysql_fetch_array qui
classe
dans un tableau les résultats ressortant de notre requête */

while($fetch=mysql_fetch_array($query))
{
    // On place en variables le résultat de chaque case du tableau
    $titre = $fetch['titre'];
    $news = StripSlashes($fetch['news']);
    /* Utilisation de la fonction StripSlashes() qui sert à supprimer les
    antislashes ajoutés
    devant les guillemets par mysql_real_escape_string() ( par précaution
)
    que l'on a utiliser auparavant dans l'ajout des enregistrements */

    /* Du fait que l'on a la date de création en format timestamp, on
    utilise la fonction date();
    qui permet de ressortir tout les renseignements voulu
    Pour plus de détails sur les formats utiliser :
    http://fr3.php.net/date */

    $date = date('d/m/Y', $fetch['date_creation']);
    $heure = date('H:i:s', $fetch['date_creation']);

    // Affichage
    echo '<strong>'.$titre.'</strong> ajouté le '.$date.' à '.$heure;
    echo $news;
    echo '<br /><br />';
}

mysql_close($dblink); // Fermeture de la connexion à la base de données
?>

```