Projet CMS

Un système de gestion de contenu ou SGC (Content Management System ou CMS) est un outil web permettant de mettre en ligne des contenus Web. WordPress, Joomla, Drupal, Spip en sont les représentants les plus populaires. Le projet consiste à concevoir une application Web permettant de générer du contenu structuré, affiché ensuite sur un site web.

Fichiers

- Sujet projet CMS V 1.0.0.2
- Base de données
- Grille d'évaluation projet

Modalités de remise du travail

- jeudi 24 avril, minuit : rendre projet PHP + Lisezmoi.txt voir Modalités de remise de votre travail
- Semaine du 12 mai : soutenances + remise rapport (pdf)

Compléments

Bonnes pratiques

- Alimenter correctement la base de données en ajoutant des enregistrements valides et en nombre suffisant, mettant en valeur votre travail
- respecter la Normalisation HTML 5/Css 3
- Structurer les fichiers et dossiers de manière cohérente
- Nommer en respectant les normes et de manière significative (pages, fonctions, variables...)
- créer et utiliser des fonctions pour :
    - Accès à la base de données pour chargements ou mises à jour d'enregistrements
    - Affichages complexes (nécessitant php) dans les pages HTML
    - Fonctions utilitaires

Exemples de fonctions

```
//Retourne une instance de connexion PDO à la base de données
function connect(){
...
}

//Retourne un tableau associatif des utilisateurs
function getUsers(){
    $users=array();
    $db = connect();
```

```
    $sql = 'SELECT * FROM user';
    $statement=$db->query($sql);
    foreach  ($statement as $row) {
        $u=array();
        $u["login"]=$row["login"];
        $u["password"]=$row["password"];
        $u["mail"]=$row["mail"];
        $users[]=$u;
    }
    return $users;
}

//Supprime l'utilisateur dont l'$id est passé en paramètre
function deleteUser($id){
...
}
```

```
//Retourne une liste HTML (<select>) des utilisateurs
function usersAsSelect($name="users"){
    $users=getUsers();
    $result="<select id='".$name."' name='".$name."'>";
    foreach ($users as $user){
        $result.="<option value='".$user["id"]."'>".$user["login"]."</option>";
    }
    $result.="</select>";
    return $result;
}

//Retourne un tableau HTML des sections
function sectionsAsTable(){
...
}
```

```
//Crée une variable de session de nom $name contenant $value
function addSessionVar($name,$value){
...
}

//Retourne les informations de l'utilisateur connecté (tableau associatif stocké en
variable de session)
function getUserInfo(){
...
}
```

SQL

Quelques exemples d'instructions SQL utilisables :

Pensez à tester votre instruction SQL dans **phpMyAdmin** avant intégration à du code PHP

--Connexion

Le mieux est de créer une fonction pour la connexion :

```
function connect(){
    $db=null;
    try{
        $db = new PDO('mysql:host=localhost;dbname=minicms',"root", "");
        $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    }catch(Exception $e){
        echo "Erreur de connexion";
    }
    return $db;
}
```

--Lecture : READ

```
SELECT * FROM SECTION;
```

```
SELECT title,description FROM section;
```

```
SELECT * FROM section where id=2;
```

```
SELECT * FROM section where idParent=2;
```

```
SELECT * FROM section INNER JOIN page ON section.idPage=page.id where section.id=2;
```

```
    $db = connect();
    $sql = 'SELECT * FROM section';
    $statement=$db->query($sql);
    foreach  ($statement as $row) {
        print $row['title'] . "\t";
        print  $row['description'] . "\t";
  }
```

--Ajout : ADD

```
INSERT INTO section VALUES('Un Titre','Aucune description',1,2);
```

```
$db = connect();
$db->exec("INSERT INTO section VALUES('Un Titre','Aucune description',1,2);");
```

-- Modification : UPDATE

```
UPDATE section SET titre='Autre Titre', Description='Autre description' WHERE id=1;
```

```
$db = connect();
$db->exec("UPDATE section SET titre='Autre Titre', Description='Autre description'
WHERE id=1;");
```

-- Suppression : DELETE

```
DELETE FROM section WHERE id=1;
```

```
$db = connect();
$db->exec("DELETE FROM section WHERE id=1;");
```

Attention, un enregistrement qui est associé à des enregistrements fils ne peut être supprimé, sans la suppression préalable de ses fils :
Exemples :

1. Si une page est associée à une section, avant de supprimer la page, il faut supprimer la section associée.
2. Si une section a des sections enfants, avant de supprimer la section, il faut supprimer les sections qui lui sont associées.

From:
http://slamwiki2.kobject.net/ - **SlamWiki 2.1**

Permanent link:
http://slamwiki2.kobject.net/php-rt/projets/projet-2014?rev=1430398107

Last update: **2019/08/31 14:26**