



Technologies du WEB

CMS & Frameworks

Tarek AIT BAHA

t.aitbaha@uiz.ac.ma

ESEF AGADIR - Université IBN ZOHR

Plan du cours



- CMS
- Frameworks

Plan du cours



- CMS
- Frameworks

CMS - Définition



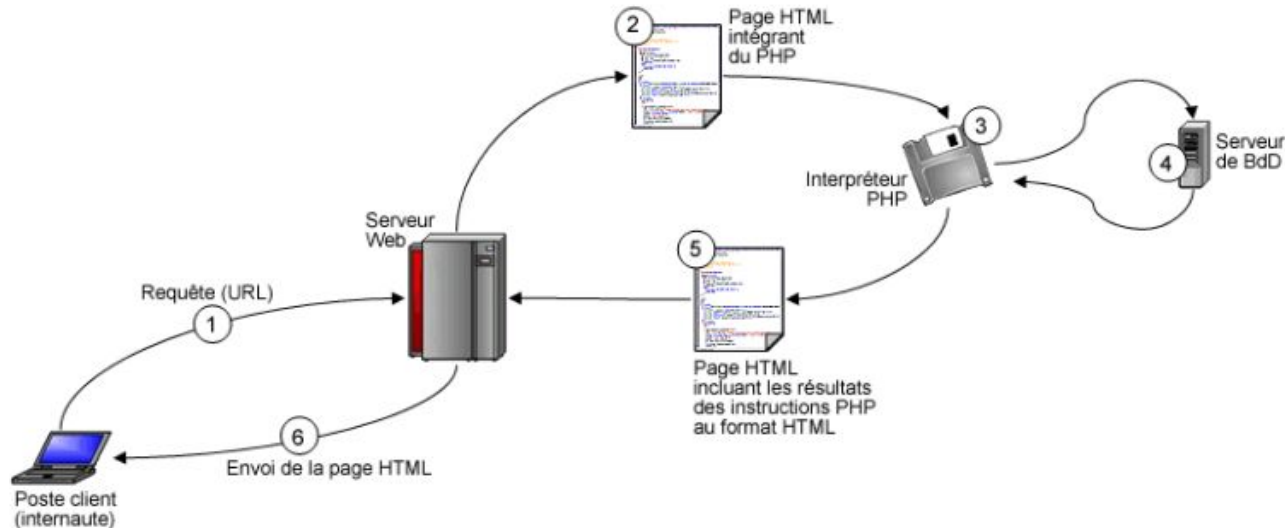
Un **système de gestion de contenu (SGC)** ou plus communément **Content Management System** pour CMS est une famille de logiciels destinés à la conception et à la mise à jour dynamique de sites Web ou d'application multimédia.

Plus précisément, il s'agit d'une application Web accessible, en général par un navigateur, qui permet de gérer le contenu d'un site Web sans avoir de connaissances informatiques poussées dans la conception de sites Web.

CMS - Fonctionnement

La plupart des CMS sont écrits à l'aide du langage PHP.

PHP est un langage s'exécutant «côté serveur» et requiert l'architecture suivante pour fonctionner :



CMS - Fonctionnement



- L'internaute effectue une requête (depuis un lien hypertexte, ou en saisissant une URL).
- La requête arrive sur le serveur web, qui transmet la page PHP à l'interpréteur PHP.
- L'interpréteur décode les instructions.
- **Si nécessaire**, l'interpréteur lance des requêtes sur le serveur de **Base de Données**.
- L'interpréteur retourne la page au serveur web. Cette page ne contient plus que du code **HTML / CSS**.
- Le serveur retourne la page à l'internaute.
- La page est interprétée par le navigateur de l'internaute.

CMS & programmation



- Il est plus facile et plus rapide d'accomplir des tâches grâce à un CMS, **exemple:** effectuer un espace membre, un forum, une boutique en ligne, etc. tout cela est possible juste en cochant une case mais il est indispensable de savoir le faire auparavant en PHP, à la main et en partant d'une page blanche.
- La raison, c'est que, à un moment donné, vous aurez certainement besoin de modifier quelque chose dans le code pour arriver à ce qu'il vous est demandé, le CMS apporte beaucoup mais même s'il apporte 90% il y a toujours du spécifique demandé par le client. Sans connaissance en langage de programmation, cela paraît difficile à accomplir.

CMS - Notion FrontOffice/BackOffice

En informatique, le back office est une partie d'un site web ou d'un système informatique.

Elle concerne la partie qui permet à l'organisation en charge d'un système informatique de l'administrer et de le gérer. On pourra retrouver comme service par exemple :

- Ajout des produits et des services
- Modification des services
- Modifications de paramètres
- Administration d'un forum
- Gestion des utilisateurs
- Création et modification des pages
- Gestion du référencement naturel –SEO

CMS - Notion FrontOffice/BackOffice

Le Back office (BO) est opposé au Front Office qui est la partie servant à l'interaction avec l'utilisateur. Avec peu de connaissances en développement, un Webmaster peut mettre à jour un site web via le back office en toute simplicité et ainsi publier de nouveaux contenus instantanément.

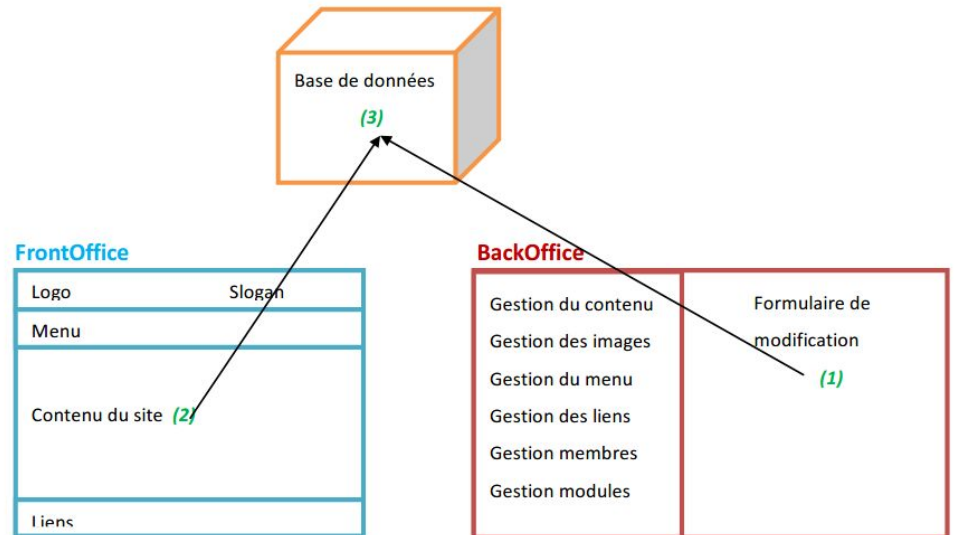
Accès:

- La partie du BackOffice étant réservée à l'administrateur, l'accès se fait via une adresse «secrète» et avec authentification requise.
- Bien qu'il soit tout à fait possible de mettre en place un FrontOffice et un BackOffice sans CMS, cette séparation est généralement présente dès l'installation du CMS.Elle est réservée au propriétaire du site ainsi qu'à son équipe d'administration.

CMS - Notion FrontOffice/BackOffice

- Nous ne pouvons pas faire une modification sur la partie BackOffice qui impacte directement le FrontOffice car il s'agit d'affichages différents, c'est la raison pour laquelle nous avons besoin d'une base de données.

- (1) Une modification se fait dans la partie BackOffice et est enregistrée en base de données.
- (2) Le Front génère son affichage en fonction des informations qu'il trouve en base de données.
- (3) Le Back et le Front ne sont pas liés directement et passent naturellement par la base de données.



CMS - Avantages liés à son utilisation

- Ces logiciels présentent de nombreux avantages dont le premier est qu'il n'est pas nécessaire de connaître les différents langages de programmation pour les mettre en œuvre.
- Ces logiciels sont gratuits et open source pour la plupart, c'est-à-dire que l'accès au code source du logiciel est autorisé par ses auteurs, donc qu'ils sont libres d'utilisation et de modifications.
- Il est possible de réaliser des sites Internet dynamiques, c'est-à-dire dont le contenu est entièrement modifiable à travers une interface de l'administration.
- Possibilité de travail simultané à plusieurs sur la même plateforme et participation valorisante des collaborateurs au contenu éditorial lié aux activités de l'entreprise.

CMS - Exemples



- [WordPress](#) est un système de gestion de contenu libre écrit en PHP et reposant sur une base de données MySQL. WordPress est surtout utilisé comme moteur de blog, mais ses fonctionnalités lui permettent également de gérer n'importe quel site web.
- [Drupal](#) est un gestionnaire de contenus(CMS) qui permet de créer une grande variété de sites Internet: blogs, portails, corporatifs, e-commerces, communautaires, réseaux sociaux, wiki, forums... Il est distribué sous licence open-source GPL. Il est donc gratuit et est développé par des milliers de développeurs expérimentés.

CMS - Exemples



- [Magento](#) est un outil puissant et technique **open source** qui permet de créer des sites **e-commerces**. Cette solution possède toute une galaxie de fonctions de base très intéressantes ainsi que des extensions gratuites ou payantes idéales pour personnaliser votre commerce en ligne.
- [PrestaShop](#) est une application Web open source permettant de créer une boutique en ligne dans le but de réaliser du commerce électronique.

Plan du cours



- CMS
- **Frameworks**

Frameworks - Définition



Un **Framework** est une sorte de cadre applicatif structurant qui permet de réduire le temps de développement des applications, tout en répondant de façon efficace aux problèmes rencontrés le plus souvent par les développeurs.

Il inclut généralement de *nombreuses fonctionnalités* prêtes à l'emploi dont les implémentations sont bien rodées et utilisent des modèles de conception standard et réputés. Le temps ainsi gagné sur les questions génériques pourra être mis à profit pour les parties spécifiques de l'application.

Frameworks - À quoi sert ?



Les framework sont utilisés afin de poser les bases génériques du programme et de disposer d'outils afin de faciliter le développement et augmenter la productivité du développeur. Il sert aussi à implémenter certain pattern (tel que le MVC ou autre...) selon les besoins et le framework choisi.

Frameworks - Pourquoi les utiliser ?



- Ils permettent de bien structurer et organiser son code.
- Avec un framework il y a toujours son lot de module et autre plugin ce qui permet un gain de temps non négligeable, ils gèrent aussi des comptes utilisateurs, de l'authentification, la connexion à différentes sources de données, etc.
- Ils vous permettent aussi de bénéficier d'une bonne base de sécurité pour le code et l'accès aux données qui sont en général pris en charge par le framework, mais aussi d'améliorer les performances et la stabilité de votre projet.

Frameworks - Comment choisir son Framework ?



Etant donné la multitude de framework qui existe il est important de bien choisir son framework avant de se lancer dans la phase de développement.

Pour bien choisir son framework il faut comprendre les besoins de son projet, en effet chaque framework a ses propres spécificités il faut donc bien prêter attention aux fonctions que le framework vous met à disposition pour ainsi que les différents patterns qu'il implémente afin de choisir celui qui répond au mieux à vos besoins et ainsi optimisera le plus votre temps de travail.

Frameworks - Exemples

Nous allons ici présenter les principaux frameworks PHP:

- [Symfony](#) : Ce framework est le plus apprécié. Il dispose d'une grande communauté, ce qui permet de trouver très facilement une solution lorsque l'on rencontre un problème lors du développement.

La modularité du framework permet de découper très facilement son projet et aide ainsi à la maintenabilité sur le long terme.

- [Laravel](#) est le principal "concurrent" de **Symfony**. Il a, lui aussi, une énorme communauté toujours prête à aider. Il permet, en plus des fonctionnalités de Symfony, d'utiliser un système d'envoi de mail, un créateur de requêtes SQL, un moteur de templates, entre autres. C'est, à titre personnel, mon favori.

Frameworks - Exemples

- Zend a la réputation d'être un framework robuste, solide, et donc parfait pour des projets d'entreprise. La communauté reste large Le framework permet aussi de protéger l'application contre un grand nombre d'attaques possibles, et c'est un énorme argument pour les projets sensibles.
- CakePHP est souvent considéré comme le plus simple à prendre en main. C'est celui qui est souvent utilisé pour entrer dans le monde des frameworks PHP. Il utilise lui aussi la convention **MVC** et reste, malgré sa simplicité, extrêmement complet.
La documentation est très facile à comprendre, et est très fournie. C'est ce qui permet, en grande partie, à ce framework d'être accessible.

Liens utiles



- [Formation Symfony](#)
- [Formation Laravel](#), [autres](#)