

Emile Cadorel
Nicolas Gaurrin
Thomas Buchet
Guillaume Dos Santos
Yoan Garnier
Cyprien Degeorge
Guillaume Gas



Cahier des charges

Projet création d'un générateur de Site Sportif

Sommaire

<u>Introduction</u>	3
<u>A. Objectifs</u>	3
<u>1. Installeur</u>	3
<u>2. Backend</u>	4
<u>3. Frontend</u>	4
<u>B. Nos outils</u>	5
<u>1. Gestionnaire de version collaboratif</u>	5
<u>2. Base de données</u>	5
<u>3. Langages</u>	5
<u>4. Moteur de template</u>	6
<u>C. Spécifications techniques</u>	6
<u>1. Structure de la base de données</u>	6
<u>2. Structure du code des différentes parties</u>	6
<u>D. Conventions de codage</u>	7
<u>1. Conventions concernant le PHP :</u>	7
<u>2. Conventions concernant le HTML et le CSS :</u>	7
<u>E. Répartition du travail</u>	8
<u>1. Installeur</u>	8
<u>2. Frontend</u>	8
<u>3. Backend</u>	8
<u>F. Améliorations possibles</u>	8
<u>G. Délais</u>	8
<u>Annexes</u>	8

Introduction

Etudiants à l'IUT d'Orléans en département informatique, un certain nombre de projets, à réaliser en équipe pour la plupart, nous ont été confié durant nos deux années d'étude. C'est ainsi que dans le cadre de ces études, on nous a donné un projet tutoré à réaliser.

Ce projet consiste en la réalisation d'un cms orienté sport dont le but est de permettre à une équipe de sport d'avoir son propre site web facilement.

Cette application doit ainsi permettre à l'équipe d'organiser divers événements, de planifier des matchs et d'afficher les résultats d'une façon claire.

Nous avons été libre de décider de l'organisation du déroulement du développement, ainsi que des outils et des langages utilisés. Nous nous sommes ainsi mis d'accord ensemble sur ces points avant de nous répartir les tâches.

Je vous propose de découvrir plus en détails ce projet à travers ce cahier des charges. Nous verrons dans un premier temps nos objectifs avant de découvrir les outils que nous avons décidé d'utiliser et pourquoi. Nous verrons ensuite les spécifications techniques de l'application, puis de quelle manière avons-nous décidé de répartir notre travail au sein de l'équipe. Enfin nous verrons quels délais nous étaient imposés ainsi que ceux que nous nous sommes imposés. Vous trouverez en annexe certains documents liés aux spécifications techniques.

A. Objectifs

1. Installateur

Le but de l'installateur est de permettre à l'utilisateur d'installer facilement son site sur son serveur. Ainsi l'installateur doit avoir une interface simple et intuitive sous forme d'étapes à suivre, et la dernière de ces étapes doit mener à la création de la base de données nécessaire au cms

L'installateur est composé de trois étapes sous forme de formulaires :

- Formulaire de connexion à la base de données. Celui-ci permet au cms d'avoir les informations nécessaires pour se connecter à la base de données de l'utilisateur afin de créer les différentes tables. L'utilisateur doit donc au préalable installer une base de données MySQL.
- Formulaire de création du compte administrateur du cms. L'utilisateur peut ainsi se connecter à la page administration avec les identifiants saisis.
- Formulaire demandant le nom du site.

Enfin, cet installateur doit permettre à l'utilisateur de visualiser un récapitulatif de toutes les données saisies et permettre de revenir sur une étape pour faire des corrections tout en gardant les données déjà saisies affichées.

2. Backend

La partie backend concerne toute l'interface permettant au créateur du cms d'y apporter des modifications. Il est le seul ayant un droit d'accès et cette partie est donc protégée par une page de login.

L'utilisateur doit pouvoir d'une part modifier l'apparence du cms, il faut lui laisser une certaine liberté au niveau des couleurs et des images par exemple, voir de la disposition de certains éléments. L'utilisateur pourra aussi choisir entre plusieurs templates disponibles, voir en créer lui-même en se basant sur la documentation programmeur.

Outre l'aspet visuel du site, son contenu doit être aussi modifiable, par exemple les news, les évènements ou les équipes doivent pouvoir être ajoutées, supprimées et modifiées. L'administrateur peut ainsi avoir une vue d'ensemble du contenu de son site et le gérer comme il le souhaite.

3. Frontend

Cette troisième partie concerne tout l'aspet visible par le public du site. L'interface doit être simple et facile à utiliser.

Voici les différentes fonctionnalités prévues :

- News (visualisation sous forme de pages, ajout de commentaires possible)
- Calendrier (regroupe tous les évènements de l'équipe)
- Equipes (permet de voir la composition de l'équipe et les détails de chaque joueur)
- Matches (possibilité de voir les différents matchs joués par l'équipe)
- Tournois (possibilité d'organiser un tournoi, de voir les scores finaux, l'évolution des matchs au fil du temps, et un export possible des scores finaux)
- Profil de l'utilisateur connecté
- Connexion/Inscription

Tous ces éléments ne doivent pas être accessibles par tout le monde, un système de droit doit permettre de donner des privilèges différents à certains utilisateurs. Enfin certaines fonctionnalités doivent être accessible que pour les utilisateurs connectés.

B. Nos outils

1. Gestionnaire de version collaboratif

Notre choix de gestionnaire de version collaboratif c'est tourné vers le logiciel git. En effet son utilisation est simple, et le site github permet de proposer des fonctionnalités intéressantes.

Tout d'abord son interface nous permet d'avoir une vision globale du projet, avec les derniers commits réalisés, la possibilité de visualiser rapidement le code source ou encore la participation de l'équipe et l'évolution du développement sous forme de graphiques. De plus github permet la création "d'issues", c'est à dire la liste des tâches à accomplir ou des bugs à corriger par exemple. Ces issues peuvent être regroupées dans des "Milestones", ce qui permet de les assigner à certaines parties de l'application. Enfin, ces milestones nous informent de l'avancement du développement au travers d'une barre de progression accompagnée d'un pourcentage de progression.

Git étant disponible à la fois sous linux et sous windows, il nous a semblé l'outil idéal pour développer notre projet en équipe.

2. Base de données

Nous avons décidé d'utiliser MySQL comme base de données. Tout d'abord parce qu'elle est gratuite, et puis pour sa simplicité. De nombreux tutoriels sont présents sur internet à son sujet et sa documentation est bien construite.

Enfin, l'utilisation de phpMyAdmin nous permet d'utiliser cette base de données encore plus simplement. C'est avec cette interface que nous pouvons effectuer des tests rapidement, par exemple des tests de contraintes entre les différentes tables. Elle permet de plus d'avoir une vue d'ensemble de la base de données et des enregistrements effectués.

3. Langages

Les langages utilisés sont actuellement HTML5 et CSS3 pour l'affichage. Ceux-ci sont en effet très répandus, et nous permettent de répondre à nos attentes assez facilement. Ils s'utilisent assez bien avec les autres langages que nous utilisons.

Nous utilisons ensuite le javascript pour certaines parties du cms, principalement pour une vérification plus poussée des formulaires du côté client. L'Ajaxy est utilisé de pair avec le javascript afin d'envoyer certaines requêtes au serveur sans avoir besoin de recharger la page. Par exemple pour le cas du champ de recherche de news avec autocomplétion.

Enfin, nous utilisons le PHP comme langage de script côté serveur afin de dialoguer avec la base de données et de faire appel à nos templates à envoyer au client. Nous avons choisi ce langage car il est aussi simple à utiliser que puissant.

4. Moteur de template

Nous avons découvert le moteur de template Smarty durant nos recherches. Cet outil codé en PHP est d'une simplicité et d'une utilité remarquable. Il nous permet, une fois les informations extraites de la base de données, de les envoyer facilement dans le template voulu avant de l'envoyer au client. Il contribue ainsi à garder une bonne organisation dans notre code, et nous permet de garder une certaine séparation entre la partie développement PHP et développement HTML/CSS.

C. Spécifications techniques

1. Structure de la base de données

Nous avons essayé d'enregistrer un maximum d'information tout en gardant une structure la plus simple possible. Ainsi, par exemple, chaque table possède un Id ce qui facilite l'accès aux différentes informations (ex : des fonctions php permettent l'accès aux données en fonction de l'Id passé en paramètre, idem pour l'insertion ou la modification de données). Vous trouverez un diagramme uml de la base de donnée en annexe.

2. Structure du code des différentes parties

La structure du code a été une étape délicate dans un premier temps mais nous pensons être arrivé à une solution pratique et bien faite.

Prenons l'exemple du frontend. Les différents éléments de celui-ci tournent autour d'un fichier index.php. Ce fichier inclut dans un premier temps vers le fichier php correspondant au "module" demandé. Par exemple dans le cas des news, c'est le fichier news.php qui sera inclus. Celui-ci exécute alors les requêtes nécessaires à la base de données avant de donner ces informations à un objet Smarty. Ce dernier appelle enfin le template correspondant aux news et l'envoie au client.

D. Conventions de codage

1. Conventions concernant le PHP :

L'indentation doit permettre une bonne lisibilité du code : une tabulation soit 4 espaces.

Pas de retour à la ligne avant un ' {' et '}' doivent rester seuls sur une ligne.

Exemple :

```
if(test) {  
    echo 'test';  
}
```

Les conditions qui n'engendrent qu'une seule ligne de code doivent tout de même avoir des '{}'.

Concernant les classes, que des caractères alphanumériques (pas de '_'). Elles doivent de plus commencer par une majuscule, tout comme le début de chaque mot composant le nom de la classe (à la manière de Java).

Les fonctions peuvent avoir un '_' entre les mots qui composent leur nom.

Les variables possèdent les même normes que le nom des classes, exception faite des variables de classe qui commencent par un 'm_' (abréviation de my).

Les constantes commencent par une '_' et un '_' sépare chacun des mots de cette constante. Elles doivent être entièrement en majuscule.

Ex : _URL_SITE

2. Conventions concernant le HTML et le CSS :

Le html soit doit avec des balises en minuscules et une indentation correcte : une tabulation (4 espaces).

Le code css à l'intérieur d'une fichier html est à proscrire.

Le code css devra respecter les mêmes normes d'appellation que le php. Quant aux accolades, les deux doivent se trouver seules sur une ligne.

E. Répartition du travail

1. Installateur

Cyprien s'est occupé de la partie base de données.
Nicolas s'est occupé de la partie HTML/CSS des différentes étapes.
Guillaume G. s'est occupé de la partie PHP de l'installateur.

2. Frontend

Développement de la partie template (HTML/CSS) : Guillaume D., César.
Développement de la partie PHP/Javascript : Emile, Guillaume G.

3. Backend

Développement de la partie HTML/CSS : Thomas, Nicolas.
Développement de la partie PHP : Yoan, Guillaume G.

F. Améliorations possibles

1. Supporter d'autres bases de données

Nous pourrions proposer à l'utilisateur lors de l'installation d'utiliser Oracle par exemple (ou autre).

2. Utiliser htmltopdf

Outil codé en php permettant la conversion d'un document html en un document pdf, peut être utile pour l'export des fiches de matchs par exemple.

G. Délais

Annexes