




RAPPORT CMS



Cardy Remy
Chef de projet : M.Melly

1 Table des matières

1	Spécifications	3
1.1	Titre.....	3
1.2	Description.....	3
1.3	Matériel et logiciels à disposition	3
1.4	Prérequis.....	3
1.5	Cahier des charges.....	3
2	Planification Initiale	3
3	Analyse.....	3
3.1	Opportunités.....	3
3.2	Document d'analyse et conception.....	3
3.3	Conception des tests	3
3.4	Planification détaillée	4
4	Réalisation.....	6
4.1	Mise en place de L'environnement	6
4.1.1	Installation du serveur web easyPHP	6
4.1.2	Téléchargement de git	6
4.2	Conception de la base de donnée	6
4.3	HTML et du CSS.....	7
4.4	Connexion à la base de données.....	8
4.5	Icones Ajout, Edition, Suppression	9
4.6	Login.....	11
4.7	Droit d'affichage	12
4.8	Création de compte	12
4.9	Mise en place de JavaScript wysiwyg	13
4.10	Récupération mot de passe oublié.....	15
4.11	Affichage des informations de l'utilisateur	16
4.12	Modifications.....	16
5	Tests	17
6	Conclusion.....	17
6.1	Bilan des fonctionnalités demandées.....	17
6.2	Bilan de la planification	18
6.3	Bilan personnel	18

7	Divers	19
7.1	Journal de travail	19
7.2	Webographie	20
8	Annexes.....	20

1 Spécifications

1.1 Titre

Universal CMS

1.2 Description

Universal CMS est un application web permettant aux utilisateurs d'ajouter eux-mêmes des pages, supprimer ou éditer.

1.3 Matériel et logiciels à disposition

Un PC standard, PHP Storm ou autre éditeur de texte, serveur EasyPhp, base MySQL, git.

1.4 Prérequis

Avoir suivi les modules 101, 133 et 151.

1.5 Cahier des charges

En annexe

2 Planification Initiale

Début du projet le 13.03.17. Celui-ci se déroule en 4 semaines et qui a pour date de fin le 7.04.17

3 Analyse

3.1 Opportunités

Ajout de menu dynamiquement

Droits des utilisateurs en fonction du type de compte

Utilisation d'une base de données pour que l'ajout se puisse faire dynamiquement des menus et articles.

3.2 Document d'analyse et conception

Documentation sur foundation, documentation sur GIT.

Tutoriels sur la confirmation par email.

3.3 Conception des tests

L'application sera testée par différents utilisateurs pour trouver les éventuelles failles de l'application.

3.4 Planification détaillée

La planification se déroule sur 4 semaines. Durant la première semaine il y a la découverte du Framework foundation, la création du html, CSS du projet, la création de la base de donnée MySQL et la connexion à celle-ci (voir figure 1).

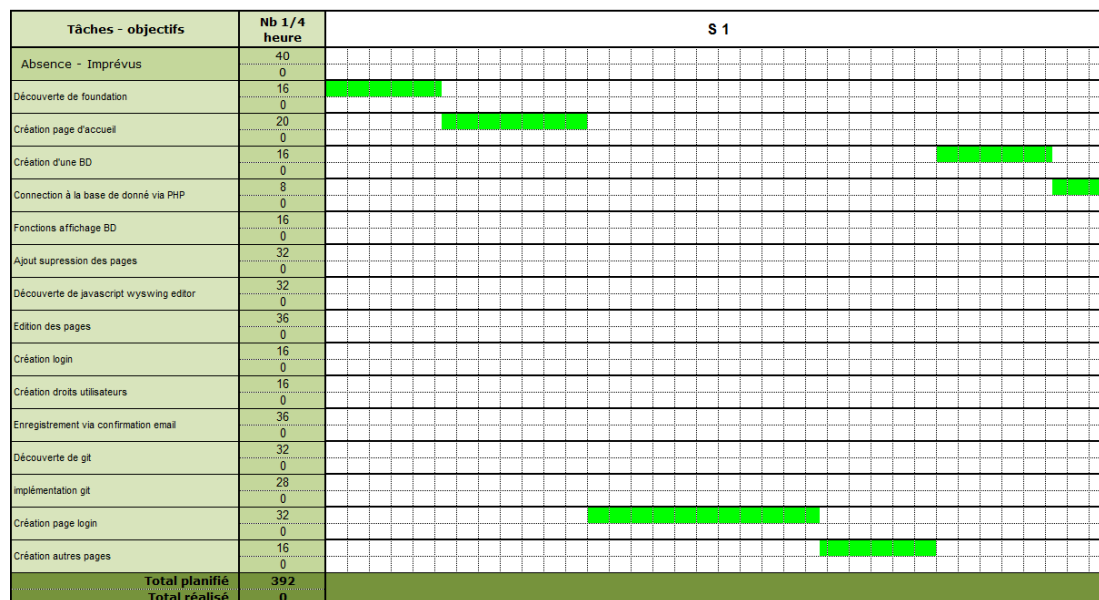


Figure 1 Planification première semaine

Durant la deuxième semaine il est prévu la création des fonctions d'affichages entre la base de donnée est PHP. Egalement il y a la création du login avec un utilisateur, la découverte et implémentation de git et la découverte de java wysiwyg (Voir figure 2).

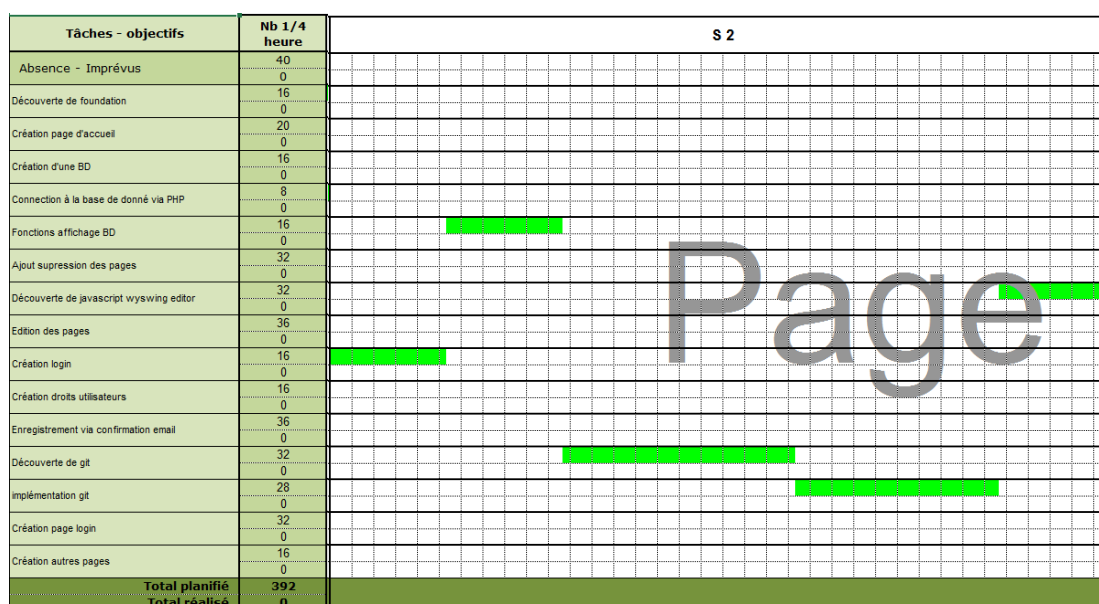


Figure 2 Planification 2ème semaine

Durant la troisième semaine, il est prévu la fonctionnalité d'ajout, édition et de suppression des pages. Il est prévu également l'ajout des droits sur les utilisateurs et l'enregistrement d'un nouvel utilisateur via une confirmation email. (Voir figure 3)

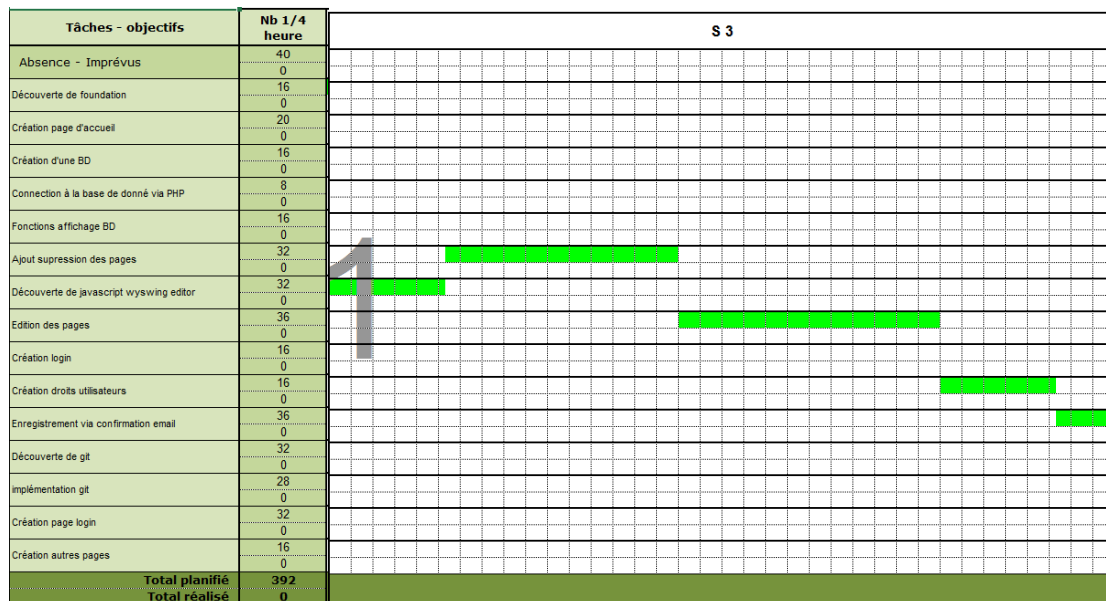


Figure 3 Planification 3ème semaine

Durant la dernière semaine il reste à finir l'enregistrement via l'email et régler tous les détails imprévus que peuvent apparaître. (Voir figure 4)

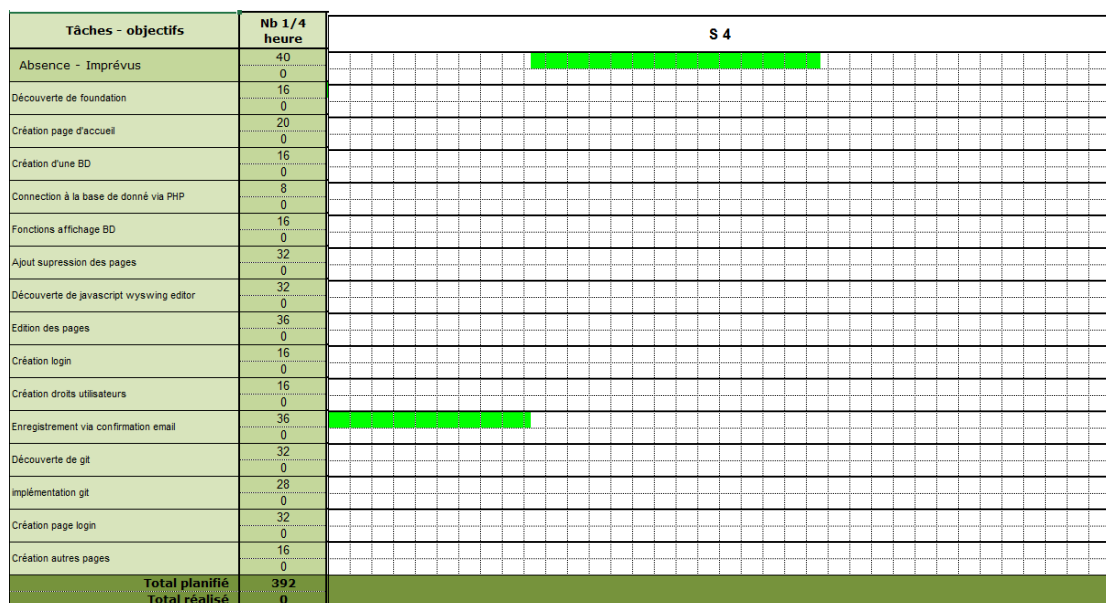


Figure 4 Planification 4ème semaine

4 Réalisation

4.1 Mise en place de L'environnement

4.1.1 Installation du serveur web easyPHP

La mise en place d'un environnement pour la création d'un CMS nécessite l'installation d'un serveur web type WAMP ou EasyPHP. Dans ce cas ça sera l'installation d'EasyPHP.

Pour télécharger easyPHP faut se rendre sur <http://www.easyphp.org>. La version utilisé est la 14.1. VC11.

4.1.2 Téléchargement de git

<https://git-scm.com/>

4.2 Conception de la base de donnée

La base de donnée est constitué de 3 tables. La table `t_user`, celle-ci contient toutes les informations liées aux utilisateurs, nom d'utilisateur, mot de passe (haché), les droits des utilisateurs, l'adresse email. Une autre table, est la table `t_article`. Celle-ci stocke les articles et les informations. La dernière table est la table `t_menu`. Cette table contient les noms des menus et elle est lié à la table articles. (Voir figure5)

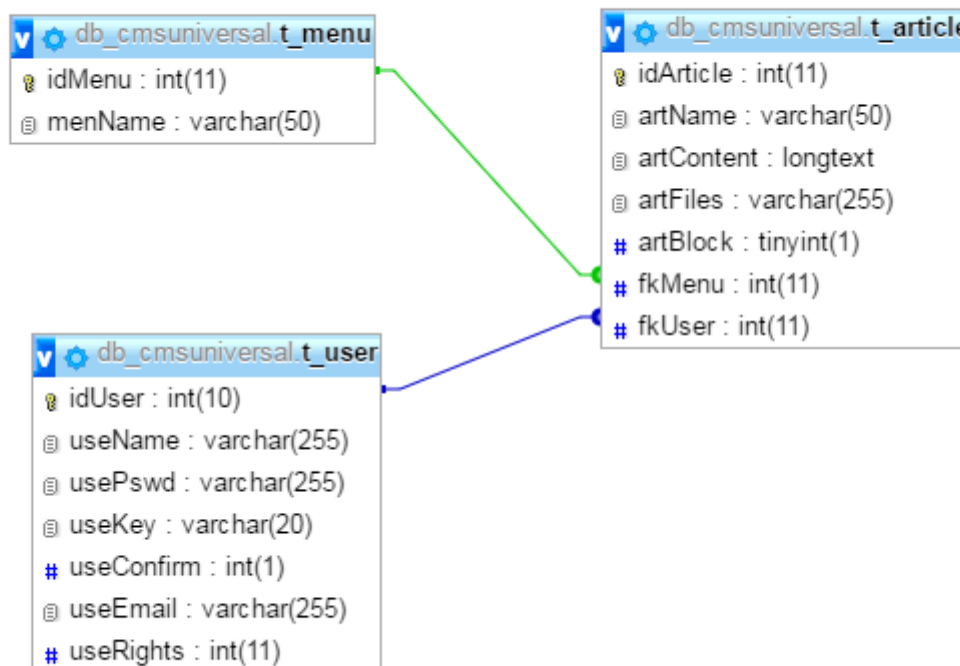


Figure 5Tables

4.3 HTML et du CSS.

Une page HTML de redirection est créée. Celle-ci sert à rediriger sur une page index de type PHP (Voir figure 6). La page index.php fait office de page d'accueil du site.

```
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Redirection en html</title>
  <meta http-equiv="refresh" content="0; URL=php/index.php">
  <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
</head>
<body>

</body>
</html>
```

Figure 6 index.html

Ensuite il y a le header qui est dans un fichier PHP. Celui-ci contient le titre, les boutons qui font offices de menus, le login et par la suite il y aura les boutons d'ajout, d'édition et de suppression. (Voir figure 7)

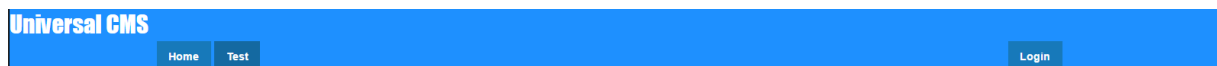
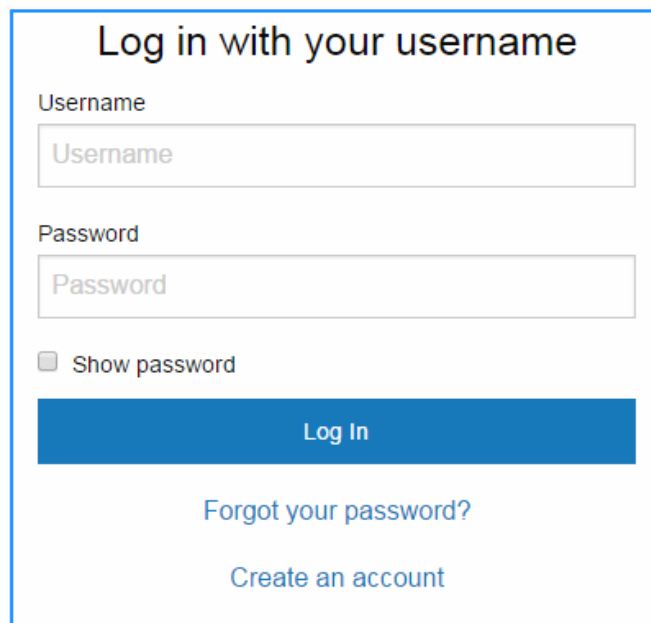


Figure 7 Header

Le footer est également dans un autre fichier PHP. Les deux sont appelés à chaque nouvelle page, ça évite de réécrire à chaque fois le code. Le footer est composé d'un lien Contact, un lien about us, un lien news et un lien login. Il est également composés d'icônes avec les réseaux sociaux.

Pour ce qui reste niveau html CSS, c'est du remplissage avec du texte et quelques images. Pour l'instant le contenu est en local et ne vient pas de la base de donnée.

La page Login est créée (Voir figure 8).



The image shows a login form with a blue border. At the top, it says 'Log in with your username'. Below this, there are two input fields: 'Username' and 'Password'. The 'Username' field has a placeholder text 'Username' and the 'Password' field has a placeholder text 'Password'. Below the 'Password' field, there is a checkbox labeled 'Show password'. At the bottom of the form, there is a blue button labeled 'Log In'. Below the button, there are two links: 'Forgot your password?' and 'Create an account'.

Figure 8 login

Elle est constituée d'un input pour l'utilisateur, un input pour le mot de passe. Il existe une fonctionnalité pour afficher le mot de passe.

Elle est en plus équipée de deux liens : un pour l'oubli du mot de passe et l'autre pour la création d'un nouvel utilisateur.

4.4 Connection à la base de données

Maintenant il faut faire la connexion à la base de données via le PHP. Pour ça il faut créer un fichier PHP, dans ce cas `function.php` qui va contenir la classe `dbFunction`. Celle-ci va contenir toutes les fonctions nécessaires pour le projet. La connexion va se faire via PDO à la base de données.

Il est possible de créer un utilisateur avec des droits spécifiques, ou d'utiliser le root. Pour créer un utilisateur il suffit d'aller dans phpMyAdmin sous utilisateurs, ajouter un utilisateur. (Voir Figure 9).

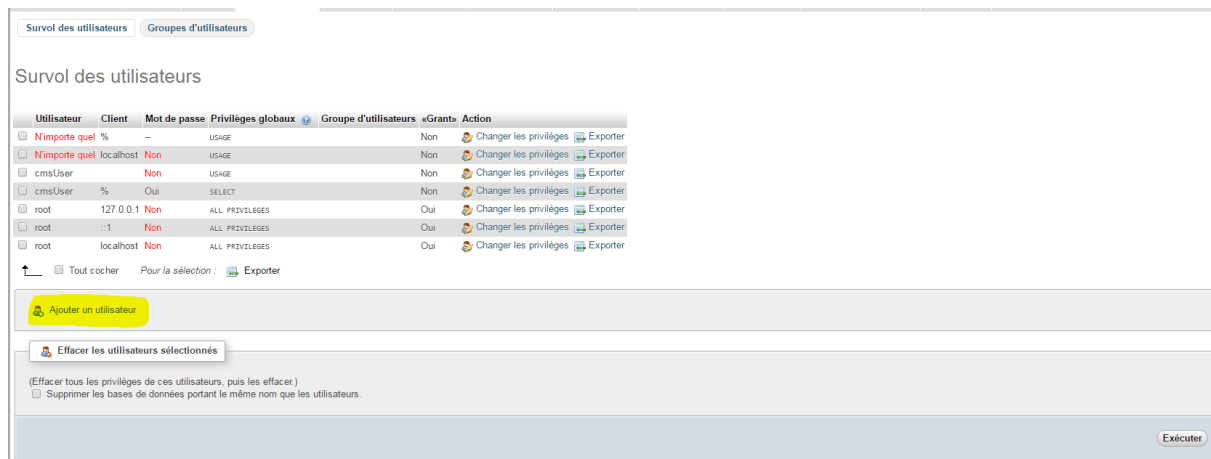


Figure 9 phpMyAdmin

Ensuite il suffit d'introduire le nom de l'utilisateur, le mot de passe et de cocher les droits sur une base de donnée définie. Cet utilisateur peut être utilisé dans le fichier PHP pour la connexion PDO.

4.5 Icones Ajout, Edition, Suppression

Une fois ceci fait, il est possible maintenant d'ajouter dans le header les icones de suppression, d'ajout et d'Édition. Ce sont des images qui vont avoir une balise <a> avec chacun un lien vers un fichier PHP (Voir figure 10).

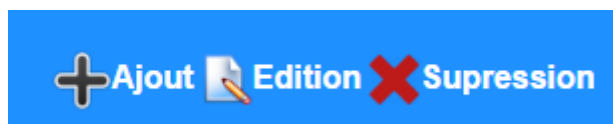


Figure 10 Icones Ajout, Edition, Suppression

Maintenant il est nécessaire de faire l'affichage des articles et des menus depuis la base de données. Pour ceci il faut créer une fonction. Pour commencer avec le header, l'affichage des boutons menus doit se faire dynamiquement. La fonction est créée dans la page fonction.php et sera appelé dans le header. La fonction se nomme sendMenuRequest est-elle est composé d'un requête SQL qui affiche toutes les informations de la table t_menu.

Pour l'affichage des menus il suffit de parcourir la base de données via une boucle for. Celle-ci va récupérer les noms des boutons. (Voir figure 11)

```
<?php
}
$countMenuData = count($loadMenuData);
//Parcourt la base de donnée
for($i=0;$i< $countMenuData;$i++)
{
?>

<a href="templateNewPages.php?id=<?php echo $loadMenuData[$i]['idMenu'];?>" class="button"><b><?php echo $loadMenuData[$i]['menName'];?></b></a>

<?php } ?>
```

Figure 11 Affichage menus

Chaque boutons aura un lien vers la page `templateNewPages.php` suivi de l'ID du menu. La page `templateNewPages` va récupérer l'id et afficher les articles en lien avec l'id du menu.

Pour l'affichages des articles, la page `templateNewPages` récupère l'id via un `$_GET` et affiche le contenu grâce à une fonction qui exécute une requête SQL.

Maintenant l'affichage peut se faire correctement.

Pour ce qui est ajout d'un menu avec des articles, il est nécessaire de créer une page `MenuAddForm.php`. Celle-ci contient un formulaire avec les champs du titre du menu, titre de l'article, du contenu et d'ajout d'une image. Cette page appelle la page `addToDB` qui elle récupère les informations comme l'id du menu, le titre du menu, le titre de l'article, le contenu et l'image. Ensuite deux fonctions sont appelées depuis la classe qui exécute les requêtes. « INSERT INTO » pour la table `t_menu` et pour la table `t_article`, permettant d'ajouter les informations à la base de donne. (Voir figure 12 pour l'insertion des articles).

```
/* *****  
 * Nom :InsertArticleInToDB  
 * But: Insert les valeurs suivantes dans la DB  
 * Retour:$getAll$getAll;  
 * Paramètre:$artName,$artContent,$artFile,$fkMenu,$user  
 * ***** */  
public function InsertArticleInToDB($artName,$artContent,$artFile,$fkMenu,$user)  
{  
    $strSQLRequestUser = "INSERT INTO t_article (artName, artContent,artFiles,fkMenu,fkUser) VALUES (?, ?, ?, ?, ?)";  
    $query = $this->objectConnection->prepare($strSQLRequestUser);  
    $rsResult = $query->execute(array($artName,$artContent,$artFile,$fkMenu,$user));  
    $getAll = $query->fetchAll();  
    $query->closeCursor();  
  
    return $getAll;  
}
```

Figure 12 Insertion DB

Pour finir dans le fichier `menuAddForm` à la fin une redirection est nécessaire vers l'index.

Une fois l'ajout fini il reste l'Edition et la suppression.

La suppression se fait grâce au fichier `deleteMenuArticle.php`, celui-ci récupère l'id du menu et va appeler deux fonctions une pour supprimer le menu et une autre pour supprimer les articles liés grâce au FK. Une redirection de la page vers l'index est mise tout à la fin.

Il reste l'Edition. Celle-ci se fait en deux parties. Le bouton Edition ramène l'utilisateur sur une page `editForm`, qui contient une forme avec les informations récupérées de la base de données grâce à la fonction `articleRequestTemplate` et une autre `getMenuDataFromID` qui ont comme paramètre l'id du menu.

Le bouton submit de la form appelle une page qui s'appelle `updateDB.php` et qui s'occupe à récupérer les valeurs du formulaire et les envoyer au fonctions de la classe qui ont comme requêtes SQL « UPDATE ».

4.6 Login

En un premier temps faut créer le login pour l'admin.

Pour le login il est nécessaire d'utiliser la fonction `password_hash`, et ensuite d'insérer le mot de passe dans la base de données pour l'utilisateur admin.

Le bouton login appelle la page `login.php` qui elle se charge de comparer si les données introduites sont les mêmes que celles de la base de données. La page `login.php` appelle deux fonctions qui affichent les informations de l'utilisateur et qui vérifie le nom d'utilisateur. Une autre étape importante est de mettre les conditions justes en cas de mauvais utilisateur, ou en cas de mauvais mot de passe.

Pour ceci c'est très simple. Avec le `$_POST` on récupère les informations du formulaire, si les champs de l'utilisateur ou du mot de passe sont vides on affiche une erreur (Voir figure 13).

```
if (isset($_POST))
{
    if (empty($_POST['strLogin']) || empty($_POST['pwd']))
    {
        $msg = "Veuillez introduire votre nom d'utilisateur et mot de passe !";
        header('Location:index.php?msg='.htmlspecialchars($msg));
    }
}
```

Figure 13 Vérification entrées

Une fonction est appelée pour vérifier l'username. La requête sql de celle-ci est d'afficher l'username de la table `t_user` dont le nom d'utilisateur = au nom d'utilisateur que l'on introduit.

Grace à ceci il suffit de poser une condition et voir si le retour de la fonction n'est pas vide, si c'est le cas un message d'erreur est généré.

```
else if(empty($userCheck[0]['userName']))
{
    $msg = "L'utilisateur n'existe pas!";
    header('Location:index.php?msg='.htmlspecialchars($msg));
}
```

Figure 14 Vérification de l'identifiant

Puis la dernière étape consiste à vérifier le mot de passe avec la fonction `password_verify`. Cette fonction compare la valeur introduite par l'utilisateur avec les données de la base de données.

```
if(password_verify($password,$userConnect[0]['usePswd']))
{
    session_start();
    //print 'Bienvenue !';
    $_SESSION['useName']= $username;
    $_SESSION['usePswd']= $userConnect[0]['usePswd'];
    $_SESSION['useRights'] = $userConnect[0]['useRights'];
    $msg = "Bienvenue";
}
else
{
    $msg="Mot de passe incorrect !";
}
```

Figure 15 Vérification du mot de passe

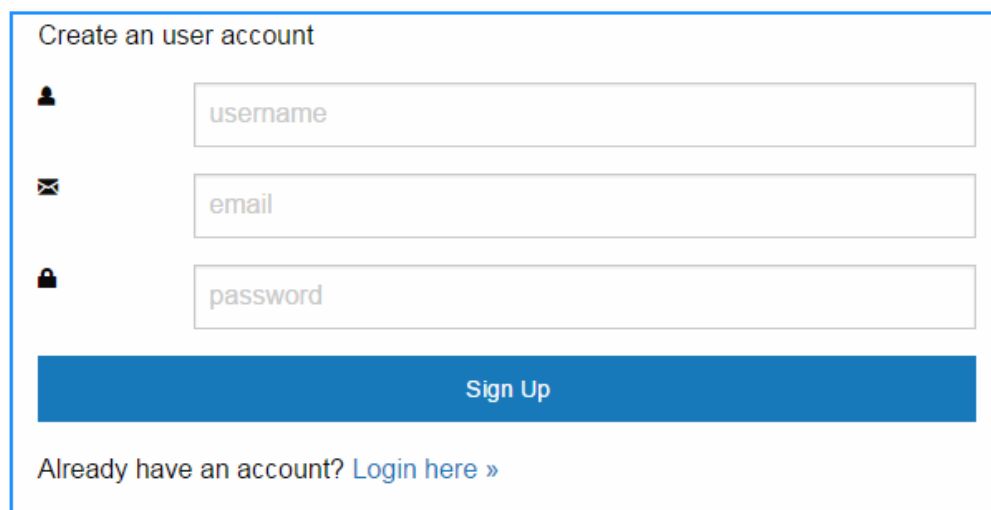
4.7 Droit d'affichage

Les droits d'affichage nécessitent une fonction qui récupère la valeur des droits de l'utilisateur qui s'est connecté. Cette valeur est inscrite dans la base de donnée.


Ensuite reste plus qu'à mettre les bonnes conditions au bons endroits. C'est-à-dire par exemple pour les icones de modification, suppression et d'ajout faut ajouter le droit 2 (le droit 2 est l'équivalent du droit admin), si l'utilisateur qui est connecté sur la session (\$_session['useRights'] == 2) à comme valeur 2 alors les boutons sont affichés sinon on n'affiche rien.


4.8 Création de compte


Pour la création d'un compte ordinaire il est utile d'avoir un formulaire avec les champs utilisateur, adresse email et mot de passe (Voir figure 16).



Create an user account







Already have an account? [Login here »](#)

Figure 16 Formulaire d'enregistrement

Ensuite l'action du bouton se fait sur la page signUP.php.

Celle-ci s'occupe de récupérer les valeurs des champs introduites. Vérifie que le format email est correct, vérifie que l'utilisateur est déjà dans la base de donnée et récupère le mot de passe.

Le mot de passe va passer d'abord dans une fonction password_hash avant d'être envoyé vers une fonction d'insertion dans la base de donnée. Grâce à une fonction javascript le mot de passe doit avoir un minimum de 6 caractères.

Pour pouvoir créer une confirmation via le mail quelques modifications sont nécessaires dans le fichier php.ini.

Le SMTP = aspmx.l.google.com (Pour les adresses gmail), smtp port = 25, sendmailfrom = me@exemple.com (ça peut être n'importe quelle adresse).

De retour dans le fichier PHP, il faut créer une clé générée aléatoirement qui sera enregistré dans la base de données et envoyé par mail en même temps.

Et pour confirmer le tout, les deux clefs sont comparées pour prouver que c'est le même utilisateur qui s'est enregistré.

Pour ceci une boucle for suffit avec deux variables. L'une \$keyLength, qui contient la longueur de la clé, ici 15 et l'autre c'est la clé elle-même grâce à la fonction mt_rand qui génère un chiffre aléatoire entre 0 et 9 et tout ceci est situé dans la boucle qui a comme longueur, la variable \$keyLength. (Pour mieux comprendre voir figure 17)

```
$keyLength = 15;
$key = '';

for($i=0;$i<$keyLength;$i++)
{
    $key.= mt_rand(0,9);
}
```

Figure 17 Création clef

Pour ce qui est de l'envoi du mail avec un lien il est utile d'utiliser la fonction mail () avec les variables \$to, \$subject, \$body, \$from. Il ne faut surtout pas oublier dans le \$from de mettre le type d'encodage comme ça le lien envoyé dans le body s'affiche sous forme de texte et non html.

4.9 Mise en place de JavaScript wysiwyg

Pour la mise en place de JavaScript wysiwyg il existe plein de solutions. L'une d'entre elle est TinyMCE. La mise en place nécessite deux étapes.

Premièrement télécharger tinyMCE sur <https://www.tinymce.com/> et deuxièmement télécharger jQuery sur <https://jquery.com/>.

Ensuite il faut créer deux dossiers. Le premier dossier sera JS, dans celui-ci il faut placer le fichier téléchargé de jQuery. Dans le deuxième dossier appelé plugins, il est important de copier tous les fichiers de tinyMCE.

Il faut également créer et lier le fichier init-tinymce.js, qui celui-ci va initialiser et déclarer les plugins pouvant être utilisés. (Voir figure 17)

```
tinymce.init({
  selector: "textarea.tinymce",
  height: 250,
  theme: 'modern',
  plugins: [
    'advlist autolink lists link image charmap print preview hr anchor pagebreak',
    'searchreplace wordcount visualblocks visualchars code fullscreen',
    'insertdatetime media nonbreaking save table contextmenu directionality',
    'emoticons template paste textcolor colorpicker textpattern imagetools codesample toc'
  ],
  toolbar1: 'undo redo | insert | styleselect | bold italic | alignleft aligncenter alignright alignjustify | bulleted numbered outdent indent | link image',
  toolbar2: 'print preview media | forecolor backcolor emoticons ',
  image_advtab: true,
  templates: [
    { title: 'Test template 1', content: 'Test 1' },
    { title: 'Test template 2', content: 'Test 2' }
  ],
  content_css: [
    '//fonts.googleapis.com/css?family=Lato:300,300i,400,400i',
    '//www.tinymce.com/css/codepen.min.css'
  ]
});
```

Figure 18 Fichier init

Maintenant pour faire le lien au fichier jQuery et au fichier tinyMCE.min.js il suffit d'ajouter des balises <script>.

Et pour faire apparaître le plugin il faut ajouter la classe : tinymce dans les inputs. (Voir figure 18 pour le résultat final.)

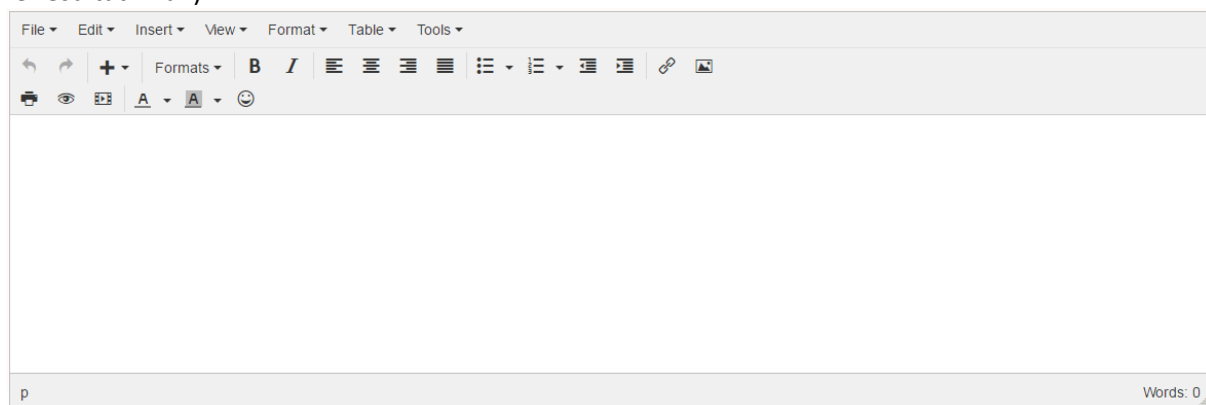


Figure 19 TinyMCE

4.10 Récupération mot de passe oublié

Le lien Forgot your Password envoie vers un champ pour introduire l'adresse email de l'utilisateur (Voir figure 20).

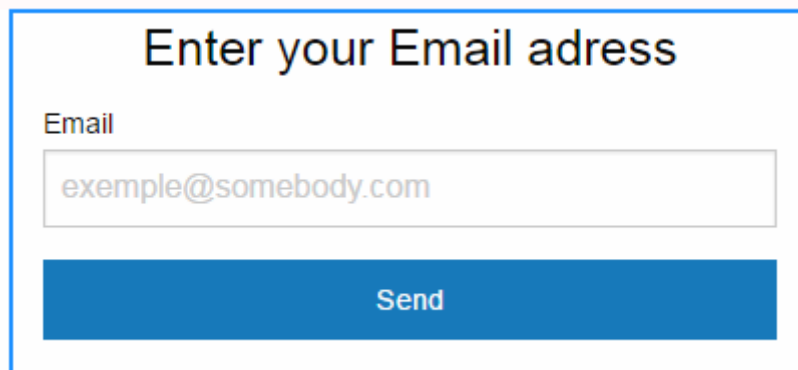


Figure 20 Forgot Your Password

Le bouton « send » envoie vers une page PHP qui s'occupe de récupérer l'adresse email, celle-ci est vérifiée pour voir si elle correspond à un format email.

La clé de l'utilisateur est récupérée de la base de données et elle est envoyée par email dans un lien, grâce à la fonction utilisée auparavant (mail()).

Lorsque le lien est cliqué, il envoie vers une page PHP qui s'occupe de vérifier de nouveau la clé, voir s'il s'agit de l'utilisateur en question. Si c'est bien le cas un formulaire pour réinitialiser le mot de passe apparaît (Voir figure 21).

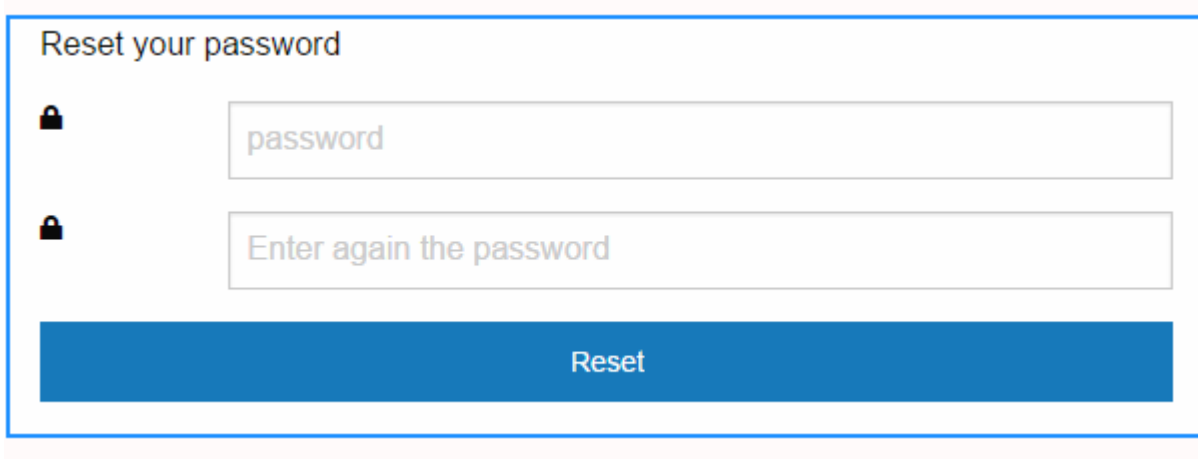


Figure 21 Reset

Celui-ci actionne une page qui va récupérer le mot de passe et le hacher et le mettre à jour dans la base de données.

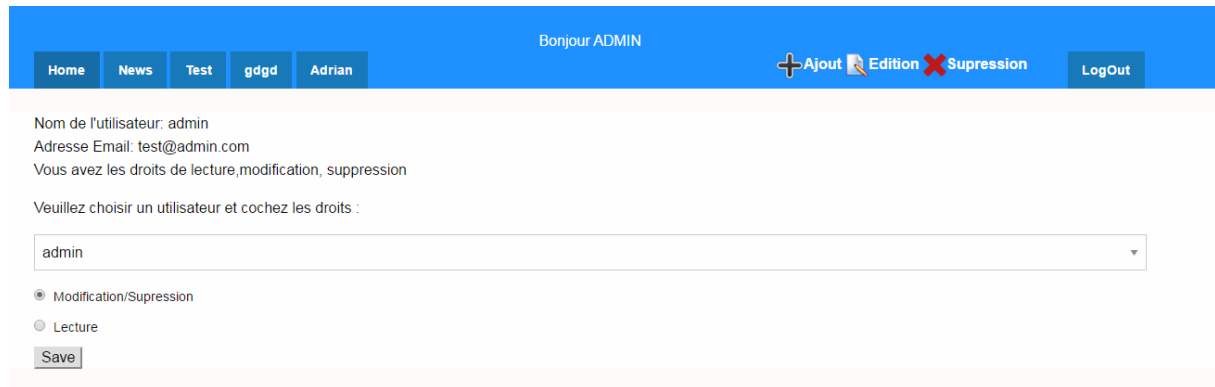
Une condition vérifie bien avant que le mot de passe a été introduit deux fois la même chose.

4.11 Affichage des informations de l'utilisateur

L'utilisateur connecté apparaît dans le header grâce à \$_SESSION. Il est possible d'afficher diverses informations lors d'un click dessus.

Une requête SQL récupère l'adresse email de l'utilisateur et les droits qu'il peut exercer.

L'utilisateur admin a quelques options supplémentaires. Il a une liste déroulante avec les utilisateurs qui s'affichent directement de la base de données. Grâce à des boutons radio il peut sélectionner les droits de chaque utilisateur (Voir figure 22).



The screenshot shows the admin interface. At the top, a blue header bar contains the text 'Bonjour ADMIN' and navigation links: 'Home', 'News', 'Test', 'gdgd', 'Adrian', '+ Ajout', 'Edition', 'Supression', and 'LogOut'. Below the header, the user information is displayed: 'Nom de l'utilisateur: admin', 'Adresse Email: test@admin.com', and 'Vous avez les droits de lecture, modification, suppression'. A dropdown menu is labeled 'Veuillez choisir un utilisateur et cochez les droits :'. The dropdown shows 'admin'. Below the dropdown, there are two radio buttons: 'Modification/Supression' (selected) and 'Lecture'. A 'Save' button is at the bottom.

Figure 22 Droits Admin

Le bouton submit envoie sur une page PHP qui prend en charge pour faire les modifications.

Elle appelle des fonctions qui vont mettre à jour la base de données pour les droits sélectionnés. Pour cela puisse se réaliser les informations sont récupérés grâce au \$_POST.

4.12 Modifications

Faire un seul même bouton d'Edition pour la page home et du reste. Le bouton Home ne vient pas de la base de donnée.

Pour ceci soit possible il est nécessaire de placer un champ supplémentaire dans la table t_articles de la base de données appelée artBlock avec une valeur de 1. Les articles ayant une valeur de 1 sont destinés à la page home.

Le choix des droits mentionné auparavant fait partie des modifications à apporter.

Le boutons d'envoi email pour récupérer le mot de passe n'est pas nommé correctement. Il est utile de ce fait de le renommer.

Le message pour confirmer la suppression d'un page n'affiche qu'elle page on est en train de supprimer. Pour modifier cela, il suffit d'ajouter un paramètre dans la fonction JavaScript.

Création de différents Template. L'utilisateur a le choix entre un menu avec un, deux ou trois articles. Celui-ci est fait grâce à une page supplémentaire qui est appelé une fois appuyé sur le bouton ajouter. Il s'agit d'une liste déroulante permettant de choisir le nombre d'articles (Voir figure 23).

Choose template



Une fois un article sélectionné il envoie sur la page menuAddForm qui s'occupe d'afficher les champs à remplir en fonction de l'option choisie.

A son tour elle envoie sur la page addToDB qui récupère les informations et insère dans la base de donnée. Quelques conditions sont nécessaires pour pouvoir insérer correctement. (Voir figure 24)

```
//Si le choix de l'utilisateur est seulement d'un article, insère les valeurs suivantes
if($templateChoose==1)
{
    $insertArticle = $objConnect->InsertArticleInToDB($articleTitle,$articleDetails,$imgRoute,$IdNumber,$idUser);
}
//Si le choix de l'utilisateur est de 2 articles, insère les valeurs suivantes
else if($templateChoose == 2)
{
    $insertArticle = $objConnect->InsertArticleInToDB($articleTitle,$articleDetails,$imgRoute,$IdNumber,$idUser);
    $insertArticle2 = $objConnect->InsertArticleInToDBWithoutfiles($articleTitle2,$articleDetails2,$IdNumber,$idUser);
    $artBlock = 2;
    $updateArtBlock= $objConnect->updateArtBlock($artBlock,$IdNumber);
}
//Si le choix de l'utilisateur est de 3 articles, insère les valeurs suivantes
else if($templateChoose==3)
{
    $insertArticle = $objConnect->InsertArticleInToDB($articleTitle,$articleDetails,$imgRoute,$IdNumber,$idUser);
    $insertArticle2 = $objConnect->InsertArticleInToDBWithoutfiles($articleTitle2,$articleDetails2,$IdNumber,$idUser);
    $insertArticle3 = $objConnect->InsertArticleInToDBWithoutfiles($articleTitle3,$articleDetails3,$IdNumber,$idUser);

    $artBlock = 3;
    $updateArtBlock= $objConnect->updateArtBlock($artBlock,$IdNumber);
}

header('Location: index.php');
```

Figure 23 Choix articles

5 Tests

Durant tout le projet des tests des fonctions sont effectués à longueur de journée.

Certains utilisateurs externes ont été interpellé pour effectuer divers tests sur l'application.

6 Conclusion

6.1 Bilan des fonctionnalités demandées

Les fonctionnalités demandées ont été créés avec succès, néanmoins il reste quelques failles de sécurité à régler sur l'application pour avoir une application 100% fiable. L'application peut cependant être déjà employé.

6.2 Bilan de la planification

La planification ne s'est pas déroulé comme prévu de base. L'ordre des opérations était différent par rapport à la planification.

6.3 Bilan personnel

Si ça serai à refaire je prendrais plus de temps pour le rapport.

Il reste encore beaucoup d'améliorations envisageables, comme par exemple de donner la possibilité à l'utilisateur de supprimer qu'un article en étant sur une page avec plusieurs articles.

7 Divers

7.1 Journal de travail

13.03.2017 – Début du projet. Mise en place de l'environnement. Installation de easyPhp. Découverte du Framework foundation. Création de la page d'accueil, header, footer, body en html et CSS.

14.03.2017 – Amélioration page d'accueil, création page login, page news, page signUp, page about us. Ajout réseaux sociaux dans le footer. Documentation sur Git.

15.03.2017 – Documentation sur git et premier commit de la semaine. Création de la Base de données avec une table, la table t_user pour tout ce qui concerne le login. Création des fonctions de connections à la BD. Création d'un utilisateur admin dans la table t_user pour pouvoir l'utiliser pour le login.

16.03.2017 - Mise à jour de la base de donnée, ajout des tables t_articles et t_menu. Affichages des boutons dans le site à partir de la base de données. Ajout des boutons « ajouter », « informations » et « modifier ». Création d'un page ajout menu et également création d'une page par défaut.

17.03.2017 – Ajout des menus possible, erreur pour le chargement des articles dedans. Création des fonctions comme Select, insert pour la BD.

20.03.2017 – Ajout des menus avec articles possible. Modification des icônes d'ajout, suppression et édition.

21.03.2017 – Résolution problème, ajout menu avec accents. Edition des articles possible mais souci lors d'une édition sans image l'image de base disparaît.

22.03.2017 – Edition des articles et du menu possible, toujours problème avec l'image. Création d'un email automatique lors du signUp, pas de confirmation encore et possible seulement sur Gmail. Edition via l'outil vyswing.

23.03.2017 – Création de la confirmation par mail. Amélioration de l'affichage.

Début de la récupération de mot de passe en cas d'oubli

24.03.2017 – Création d'un nouveau mot de passe après une confirmation par email possible. Commit ;

27.03.2017 – Affichage de la page home depuis la BD et création de la page d'Édition pour celle-ci.

28.03.2017 –Affichage des informations utilisateurs sur le site et mise en place d'ajout de droits aux autres utilisateurs si connectés en admin.

3.04.2017 – Modifications apportés au code pour améliorer l'application CMS

4.04.2017 – Rédaction rapport

5.04.2017 – Correction de certains bugs

6.04.2017 – Création des différents Template, insertion possible avec un, deux ou trois articles

7.04.2017 – Suppression possible des articles après modification avec Template et édition possible des articles.

7.2 Webographie

<http://www.easyphp.org/>

<http://foundation.zurb.com/>

<https://git-scm.com/video/quick-wins>

<https://www.youtube.com/watch?v=V6Zo68uQPqE>

<https://www.pexels.com/>

<http://stackoverflow.com/questions/10904098/why-text-column-returns-only-4096-bytes>

8 Annexes

Cahier des charges