Projet DemoMot



Timothée Guggisberg

ETML

4 Semaines

Mr Gilbert Gruaz et Mr Laurent Deschamps

Table des matières

[1 Introduction 3](#_Toc485027436)

[1.1 Explication du projet 3](#_Toc485027437)

[1.2 Méthode de projet 3](#_Toc485027438)

[2 Analyse 3](#_Toc485027439)

[2.1 Réalisation/Mise en place du HTML 3](#_Toc485027440)

[2.2 Base de donnée 3](#_Toc485027441)

[2.2.1 MCD 3](#_Toc485027442)

[2.2.2 MLD 4](#_Toc485027443)

[2.2.3 MPD 5](#_Toc485027444)

[2.3 Structure du code 6](#_Toc485027445)

[2.3.1 Découpe du code 6](#_Toc485027446)

[2.3.2 Utilisation du Framework 6](#_Toc485027447)

[3 Réalisation 8](#_Toc485027448)

[3.1 Manuel d’utilisation (admin) 8](#_Toc485027449)

[3.2 Manuel d’utilisation (utilisateur) 9](#_Toc485027450)

[3.3 Calendrier 10](#_Toc485027451)

[4 Tests 10](#_Toc485027452)

[4.1 Test réalisé 10](#_Toc485027453)

[5 Conclusion 11](#_Toc485027454)

[5.1 Conclusion générale 11](#_Toc485027455)

[5.2 Conclusion personnelle 11](#_Toc485027456)

[5.3 Conclusion de la planification 11](#_Toc485027457)

[6 Divers 11](#_Toc485027458)

[6.1 Webographie 11](#_Toc485027459)

# Introduction

## Explication du projet

Le projet est un site web connecté à une base de donnée MySQL afin d’y stocker les données. Le site sert à réserver des plages horaires de mon chalet, il permet d’y avoir plus d’organisation de la gestion des réservations du chalet. Le site possède donc un calendrier avec les plages horaires confirmée par les administrateurs du site web, il y en a donc 5, un pour chaque famille à qui appartient le chalet ainsi qu’un pour moi-même pour la gestion du site et de la base de données en elle-même.

Du côté de la base de données, elle est donc gérée ave PhpMyAdmin avec un paquet de serveurs WAMP.

Le projet est sur 5 semaines dont 1 étant pendant le camp de sport et une ayant que la présentation le mardi 28 Juin. Cela fait donc 3 semaines de production du site, de la base de données et de la documentation avec la théorie un jour et demi par semaine.

## Méthode de projet

Le projet est planifié en cascade, donc la planification initiale a été faite le premier jour du projet plutôt que tout au long du projet tel qu’une méthode AGILE. La planification est sur Excel avec le modèle du journal de travail fournit par l’ETML.

# Analyse

## Réalisation/Mise en place du HTML

L’HTML du site a été pris d’un Framework CSS nommé Materialize, le site est découpé en 4 parties, une page nav.php, head.php, footer.php et le body étant dans chaque page php

## Base de donnée

La base de données a été modélisée avec JMerise puis gérée avec PhpMyAdmin, elle possède 2 version car une table a été implémentée par la suite du projet, la table event comportant toutes les plages horaires acceptée par un administrateur, elle utilise le calendrier pour fonctionner et donc est quelque peu spéciale par rapport aux autres tables

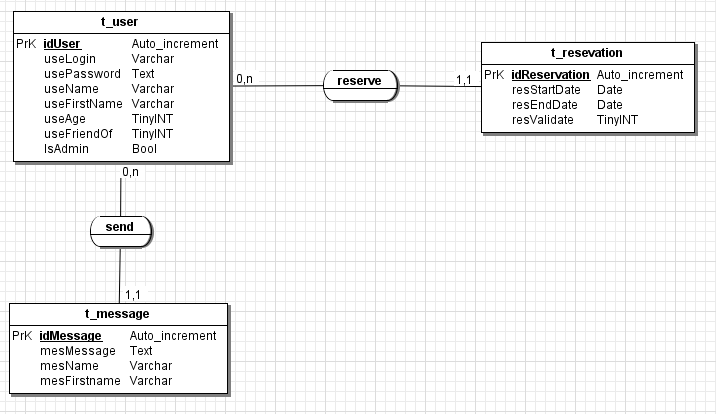
### MCD

Le MCD comporte donc 3 table :

t\_user : comportant donc les utilisateurs et les administrateurs du site web avec d’un côté les informations personnelles (Nom, prénom, age, etc..) et de l’autre les identifiants de connexion (Login et mot de passe)

t\_reservation : comportant les demandes de plage horaires des utilisateurs avec une date de début et une date de fin

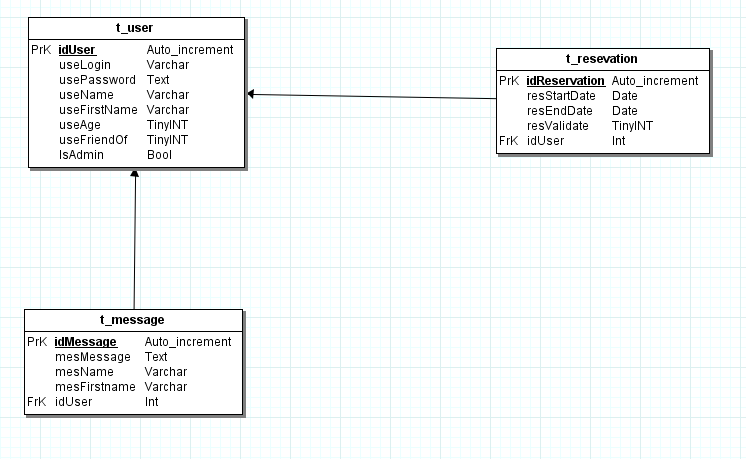
t\_message : comportant les messages envoyés par les utilisateurs en cas de questions pour les administrateurs



L’utilisateur est donc la table centrale pouvant donc faire 0 ou plusieurs réservation et 0 ou plusieurs messages, les messages et les réservations appartienne quant à elles qu’à un seul utilisateur

### MLD

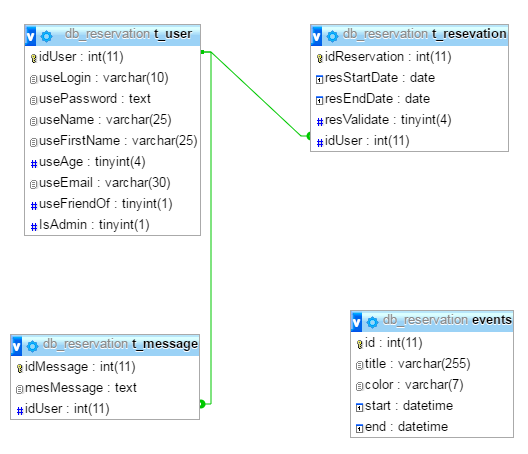
En passant le MCD en MLD, aucune autre table ne se crée et toutes les clés étrangères sont des idUser, t\_user étant la table centrale du site



### MPD

C’est donc dans cette partie-là que la base de donnée a changé après une semaine environs, en y ajoutant la table events, voici donc le MLD dans PhpMyAdmin de la nouvelle base de donnée

La table event n’est donc connectée à aucune autre table mais a pourtant un rapport avec certaines. Effectivement, la table events est l’équivalent de la table réservation une fois acceptée par un administrateur, le transfère se fait depuis le site et non la base de donnée, la table events étant donc donnée avec la libraire Full Calendar afin d’avoir un bon calendrier géré en JavaScript



## Structure du code

### C:\Users\guggisbeti\Desktop\Capture.PNGDécoupe du code

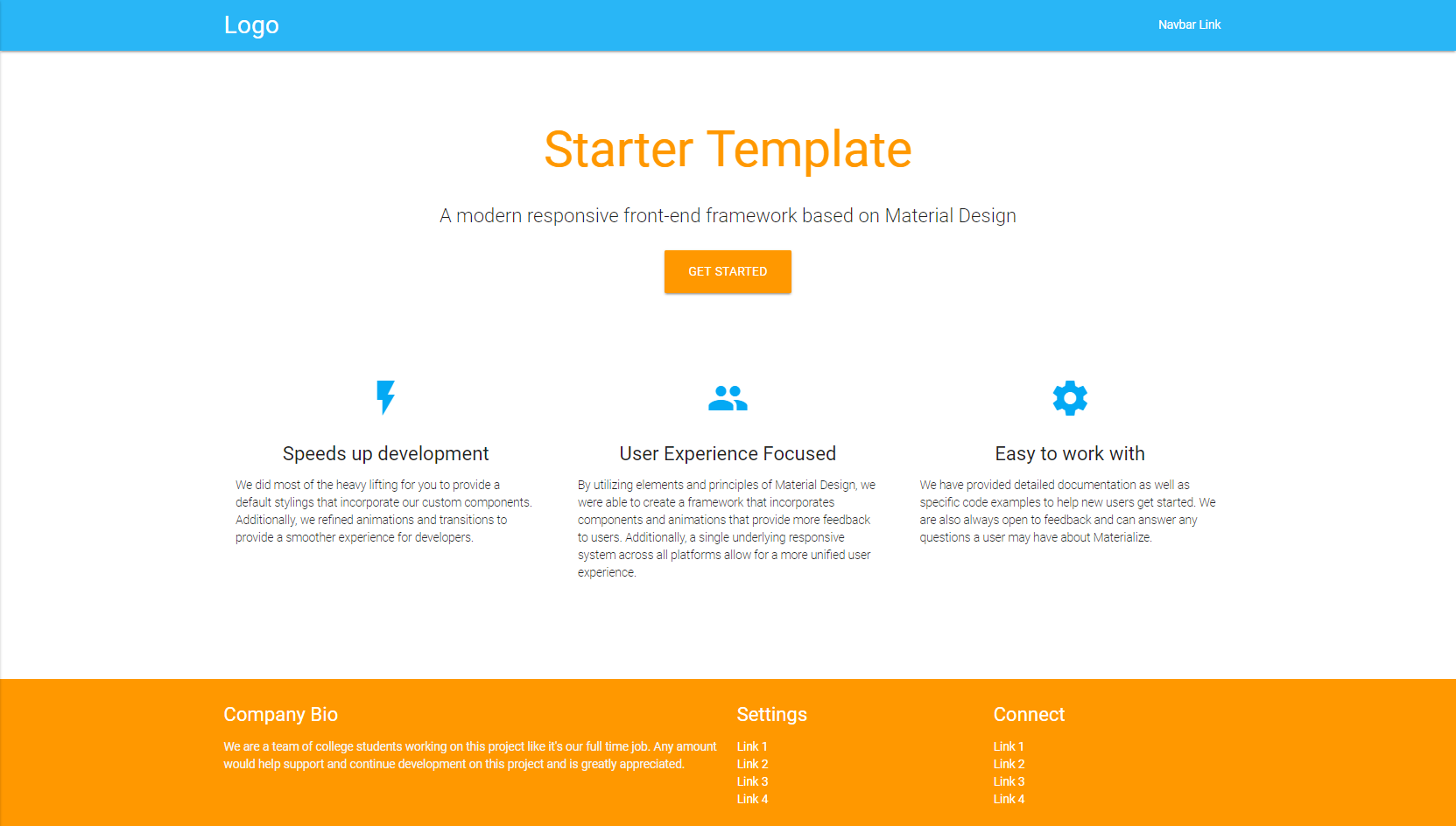
Voici toutes les pages du site web avec dans l’inclue les parties à inclue dans la quasi-totalité des pages, possédant donc la navigation, le footer, le head et la classe PDOLink pour les connexion PDO avec la base de données, la page config.php est utilisée afin de changer plus facilement les données pour la connexion à la base de données (Le login, le mot de passe, le nom de la base de données)

Les autres pages sont dans le dossier PHP avec toutes les pages utilisées dans le site web, même si certaines ne se voient pas ayant donc que du php à l’intérieur, comme la page de déconnexion resetant simplement les sessions avec une redirection sur la page d’accueil

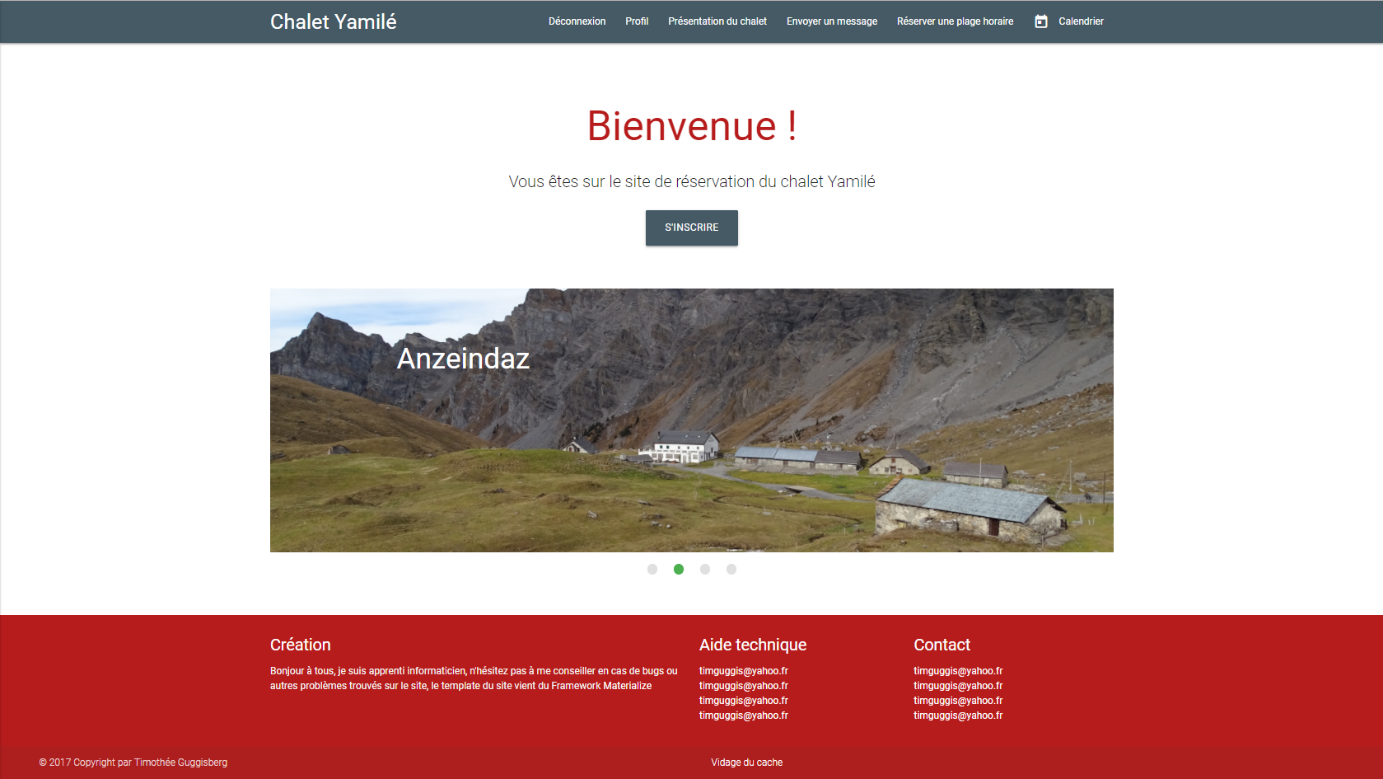
### Utilisation du Framework

Le Framework utilisé est Materialize, offrant plusieurs composant JavaScript et CSS afin d’améliorer le site web au niveau du design. L’utilisation de celui-ci c’est alors passé durant tout le projet afin de toujours rester dans le même type de design. Les couleurs principales du site sont le bleu-gris et le rouge foncé, ce sont les couleurs du blason de ma famille, le site en étant dédié, j’ai décidé de prendre ces couleurs pour le site.

Le Framework Materialize utilise aussi du JQuery dans son code pour toutes les parties JavaScript, il est utilisé par exemple dans le Slider de la page d’accueil (Voir ci-dessous)

[](http://materializecss.com/templates/starter-template/preview.html)Voici donc la page proposée par le Framework :

C’était la première fois que je devais utiliser un Framework, ne connaissant peu dans le sujet, j’ai donc pris un ensemble directement afin de m’approprier le Framework

Voici donc la page d’accueil de mon site web :

J’ai donc changé les couleurs et ajouter un Slider (composant Materialize permettant d’afficher des images) j’ai tout de même garder toute la disposition proposée par Materialize

# Réalisation

## Manuel d’utilisation (admin)

**CONNEXION :**

La manière pour devenir administrateur se fait seulement depuis la base de données (Etant une donnée fixe avec seulement 5 admins) pour plus de sécurité. L’utilisateur doit alors créer son compte je dois alors le passer en admin dans la base de données (passer isAdmin de 0 à 1) après cela il devra se connecter avec le bouton de connexion. Il peut alors mettre son login et son mot de passe fraichement créé dans la page d’inscription. La redirection va alors l’envoyer sur la page d’accueil de l’administrateur, du moins si le login et le mot de passe match avec ceux de la base de donnée.

C:\Users\guggisbeti\Desktop\Capture.PNGLa nav va donc changer avec les fonctionnalités admises au administrateurs

**PROFILE :**

La page profile mène sur une carte montrant les données insérées depuis l’inscription, un bouton Editer permet de changer les données, tel que le prénom, le nom, l’âge mais les identifiants pour la connexion, la carte a aussi un autre raccourci vers le calendrier avec le bouton Réserver une plage horaire

**PRÉSENTATION DU CHALET :**

Cette page montre simplement quelques photos du chalet avec quelques informations sur la gestion de l’eau et du chauffage là-bas cette page est accessible par les admin et les utilisateurs

**MESSAGE :**

Cette page est réservée au administrateur, regroupant donc tous les messages des utilisateurs normaux qui quant à eux peuvent envoyer des messages au administrateurs, les messages sont sous forme d’une carte avec deux boutons, un pour les informations de l’utilisateur qui l’as envoyé tel que le nom, le prénom et l’e-mail si besoin de lui répondre, cette page a aussi un bouton supprimer afin de supprimer le message. L’autre bouton est le message envoyé par l’utilisateur.

**PLAGE HORAIRES DEMANDÉE :**

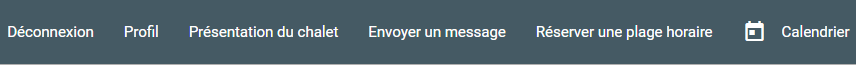
Plus au moins le même type que les messages, les utilisateurs peuvent demander des plages horaires pour les administrateurs qu’ils peuvent ensuite ajouter au calendrier dynamiquement ils peuvent l’ajouter ou le refuser, s’ils le refusent la plage horaire se supprimer automatiquement

**CALENDRIER :**

Le calendrier est la page principale du site, elle gère toutes les plages horaires du site étant alors acceptée par les administrateurs, lorsque l’on va sur le calendrier en tant qu’administrateur, on peut ajouter des plages horaires en sélectionnant simplement la durée de la réservation, on y choisi alors une couleur suivant l’administrateur (Une couleur pour chaque admin) ainsi qu’un titre, les dates sont automatiquement mises suivant les dates sélectionnées. Elles peuvent aussi être modifier en double cliquant la plage horaire. Ils arrivent alors sur le même formulaire qu’avant mais peuvent changer le titre et la couleur, on peut aussi l’agrandir, la réduire ou encore la déplacer, en prenant simplement la plage horaire et la déplacement en maintenant la souris dessus, pour agrandir et réduire il faut prendre la plage horaire à son extrémité droite et déplacer comme l’on veut.

## Manuel d’utilisation (utilisateur)

**CONNEXION :**

La première fois sur le site, il faut s’inscrire en y ajoutant les identifiants du login et du mot de passe, une fois cette étape faite, l’utilisateur est dans la base de données et peut donc se connecter avec l’onglet connexion, une fois connecté, la barre de navigation va changer avec les fonctionnalités admises au utilisateurs

**PROFILE :**La page profile mène sur une carte montrant les données insérées depuis l’inscription, un bouton Editer permet de changer les données, tel que le prénom, le nom, l’âge mais les identifiants pour la connexion, la carte a aussi un autre raccourci vers les réservation de plage horaire avec le bouton Réserver une plage horaire

**PRÉSENTATION DU CHALET :**

Cette page montre simplement quelques photos du chalet avec quelques informations sur la gestion de l’eau et du chauffage là-bas cette page est accessible par les admin et les utilisateurs

**ENVOYER UN MESSAGE :**

Cette page envoi sur un formulaire avec un champ message pouvant envoyer un message au administrateur dans un onglet réserver, les messages peut avoir maximum 600 caractères avec un compteur durant l’écriture du message, une fois envoyé il arrive dans la base de données et les administrateurs peuvent alors lire le message et vous répondre par votre adresse e-mail

**RÉSERVER UNE PLAGE HORAIRE :**

Cette page est un simple formulaire avec deux champs pour des dates, une pour la date de début et l’autre pour la date de fin, celle-ci s’envoi sur une page réservée aux administrateurs pouvant accepter votre demande, si elle est acceptée, elle va alors s’insérer dans le calendrier avec les autres plages horaires acceptée par les administrateurs.

**CALENDRIER :**

La page du calendrier est la même que pour les admins mais vous ne pouvez que voir les dates réservées afin de vous renseigner sur quand mettre votre réservation, les admins ont plusieurs droits comme insérer, modifier ou supprimer mais les utilisateurs normaux sont en lecture sur la page est ne peuvent rien modifier.

## Calendrier

Le calendrier a été pris d’une bibliothèque JavaScript afin d’y avoir plus d’automatisme est de design malheureusement, elle utilise un peu de bootstrap pour le design, j’ai donc modifier un peu le design sans détruire le calendrier ce qui change un peu la police lorsque nous sommes sur la page du calendrier

C:\Users\guggisbeti\Desktop\Capture1.PNGC:\Users\guggisbeti\Desktop\Capture.PNGC:\Users\guggisbeti\Desktop\Capture.PNGC:\Users\guggisbeti\Desktop\Capture1.PNG Materialize : Calendrier :

Mais les changements entre les deux restes alors minime.

# Tests

## Test réalisé

**Connexion à la base de donnée :**

Afin de vérifier la connexion avec la base de donnée depuis le PHP il existe plusieurs manières :

1. Tout d’abord vérifier si la requête liée à la connexion est fonctionnelle, pour cela je faisais un simple « echo $query » $query étant donc la requête, une fois cela fait j’allais sur la page web et copiais la requête pour la coller sur le SQL de phpMyAdmin puis lancais simplement la requête, si celle-ci fonctionnait, le problème ne venait donc pas de la requête.
2. Une autre manière était juste de tester le formulaire en lui-même, en remplissant tous les champs et lisant les erreurs ou si la base de donnée avait bien été remplie
3. Si ces deux points étaient testés, la dernière chose à faire était de relire le code jusqu’à trouver une erreur ou encore recommencer le code depuis le départ puis comparer les deux fichiers entre eux

**Vérification des expressions régulière :**

Les regex ont été codé dans mon site en directement sur le PHP, tester ces regex était tout d’abord pendant leur création sur le site : <https://regex101.com/>, un site dans lequel on peut écrire des regex puis les tester en direct

Suite à cela, il fallait simplement vérifier que les regex fasse le travail en vérifiant les champs du formulaire, en y mettant des valeurs fausse par rapport aux regex

**Suppression :**

Pour les tests sur la suppression, une petite erreur peut être fatale pour la base de données en supprimant simplement toutes les données d’une table, il ne faut donc pas faire d’erreur durant un delete, pour ce faire je testais la requête depuis une autre base de données (la même mais sur un autre serveur) mais surtout y mettre un WHERE afin d’être quasi sûr que la requête ne va pas tout supprimer

**Modification :**

La modification a aussi son lot de problème du même type que la suppression car plutôt que tout supprimer, celle-ci va changer les valeurs de toutes les données si la requête est mal faite, j’ai donc opté pour la même technique que la suppression en allant sur une autre base de données même si aucun problème de ce type n’est survenu

**Sécurité :**

URL :

1.

J’ai ajouté un if au début de chaque pages suite à des tests effectués dans lesquels je mettais directement l’URL des pages sans être connecté, je pouvais alors me balader sur le site sans y être connecté, et même sur les pages des administrateurs, j’ai donc mis une vérification qui va dire que si l’utilisateur n’est pas connecté : redirection sur la page d’accueil instantanée, et sur les pages admin un autre if prenant en compte la session des administrateurs, si l’utilisateur n’est pas un admin et tente d’aller sur une page administrateur, même chose, une redirection sur la page d’accueil instantanée

2.

Afin de ne pas avoir d’erreur lorsque que quelqu’un changeait l’URL dans une page prenant des éléments de l’URL (Ex : inscription?type=inscription) lorsque qu’un GET prenait une valeur, un if était ajouté afin de vérifier si les champs de l’URL existaient s’il n’existaient pas, ne prenait aucun GET afin de ne pas avoir d’erreur de type undefined index

TAMPER DATA :

J’ai aussi installé un plug-in (Tamper data, utilisable sur FireFox) permettant alors de changer les données d’un formulaire après le JavaScript, mais mon site prenait déjà en compte cela et m’as donc été peu utile mais tout de même important à faire si l’on ne veut pas se faire hacker la base de données

# Conclusion

## Conclusion générale

Le projet s’est bien déroulé et tous les objectifs du projet ont été atteint à mon gout, le site fonctionne comme voulu et le design me plait beaucoup, je n’ai pas eu beaucoup de problème ou du moins pas pendant tout le projet, même si parfois il m’a fallu plus d’une journée pour le corrigé j’ai réussi à régler tous mes problèmes dans le temps imparti.

La planification m’a semblé par contre peu correcte par rapport au travail fait chaque jour, je me suis souvent mis trop de temps pour certaines choses et quelque fois pas assez mais dans l’ensemble le nombre de périodes est correct.

L’utilisation de GIT n’était pas indispensable mais c’est toujours un plus pour partager ses données avec le professeur et pour avoir un dossier propre de mon DemoMot, j’avais simplement la branche Master avec la documentation et le travail fini, puis une branche site et une Base de données

## Conclusion personnelle

Ce projet est pour moi une réussite, j’ai réussi à faire le travail demandé et aussi ajouter plusieurs fonctionnalités à mon site web, j’ai mis un maximum de sécurité sur le site web en tentant du mieux que je peux de casser mon site pour ensuite corriger les failles.

L’année passée j’avais fait mon DemoMot avec Raphael et c’était donc intéressant de faire celui-ci tout seul afin de m’entrainer pour le TPI, ayant beaucoup de peine à faire les choses seuls sans ne jamais être sûr que ce que je fais est donc correct, c’était donc une expérience très enrichissante pour moi qui m’as permis de m’entrainer à combattre ce problème que j’ai, tout en faisant un maximum pour être le plus autonome sans demander à chaque problème l’aide de quelqu’un.

Malgré le camp qui nous enlève une semaine de DemoMot j’ai trouvé le temps donné parfait pour le TPI étant plus au moins équivalent à celui-ci

# Divers

## Webographie

Format de date dd.mm.yyyy à yyyy-mm-dd :

<https://css-tricks.com/snippets/php/change-date-from-ddmmyyyy-to-yyyy-dd-mm/>

Intégration du SQL dans Full Calendar :

<http://jamelbaz.com/tutos/integration-de-fullcalendar2-php-mysql>

Materialize :

<http://materializecss.com/templates/starter-template/preview.html>

Format de date JS :

<https://codepen.io/anon/pen/oXEypq>

Php.net :

<http://php.net/manual/fr/book.mail.php>

Mailto :

<http://www.thelin.net/laurent/labo/html/mailto.html>

Regex :

<https://regex101.com/>

Redirection :

<https://stackoverflow.com/questions/503093/how-to-redirect-to-another-webpage-in-javascript-jquery>