



# TPI 2018

# KIDSGAMES SCORES

CONSTANTIN HERRMANN

ENSEIGNANT: PASCAL BONVIN

## MANUEL TECHNIQUE



<b>Introduction</b>	<b>5</b>
Les Kidsgames	5
Mon projet	5
Résumé	5
Pourquoi j'ai choisi ce projet	5
<b>Cahier des charges</b>	<b>6</b>
Objectif	6
Fonctionnalités à réaliser	6
Travail à rendre	6
Planning	7
<b>Analyse fonctionnelle</b>	<b>7</b>
Description globale	7
Description détaillée	7
Description de l'interface	8
Le template	8
Présentation de l'interface utilisateur	8
La barre de navigation	10
Page d'accueil	10
Page classement	11
Page équipe	12
Page staff	12
Page nouveau match	13
Page du jour	14
Page match	15
Page modification du score	15
Page du planning d'une équipe	16
Page de mes matchs	16
<b>Analyse organique</b>	<b>17</b>
Architecture	17
Base de données	17
Informations	17
Accès	17
MCD	17
Sécurité des vues	18
Page nouveau match	18
Page match	18
Page modification du score	19
Page planning d'une équipe	19
Page de mes matchs	19
Script de suppression d'un membre du staff	19
Script de suppression d'un match	19
Page administration des membres du staff	19
Méthode de codage	20

Prepare statement	20
Noms et descriptions	20
En-tête des fichiers	21
<b>Interactions avec l'utilisateur</b>	<b>21</b>
Mauvais mot de passe ou identifiant	21
Réinitialisation du camp	21
Modification Et Ajout d'un membre du staff	22
Creation d'un match	22
<b>Fonctionnalités implicites</b>	<b>22</b>
<b>Descriptions des fonctionnalités</b>	<b>22</b>
Description de la fonction login	22
Description de la fonction ResetCamp	23
Description de la fonction GetClassement	23
Description de la fonction GetDays	24
Description de la fonction GetMatchsOnDayAndTime	24
Description de la fonction GetDayById	24
Description de la fonction GetFields	24
Description de la fonction GetFieldById	25
Description de la fonction CreateNewGame	25
Description de la fonction GetAllmatchsInfos	25
Description de la fonction IsArbitreOrAdmin	26
Description de la fonction CountGamesOnDay	26
Description de la fonction DeleteGame	27
Description de la fonction GetMatchsForArbitreDayAndTime	27
Description de la fonction UpdateScore	27
Description de la fonction getMainClassement	28
Description de la fonction GetMatchResults	29
Description de la fonction MakeTheResults	29
Description de la fonction MakeThePoints	30
Description de la fonction UpdateMainTable	30
Description de la fonction GetTypeOfClassement	31
Description de la fonction SetGameToDo	31
Description de la fonction GetScoreForTeamInMatch	31
Description de la fonction GetRoleById	31
Description de la fonction GetAllRoles	32
Description de la fonction GetSportTypes	32
Description de la fonction GetSportById	32
Description de la fonction GetIdsByPlayerCount	32
Description de la fonction GetStaffById	33
Description de la fonction GetAllStaff	33
Description de la fonction AddStaff	34
Description de la fonction UpdateStaff	34
Description de la fonction DeleteStaff	34
Description de la fonction GetAllCoachsWithoutTeam	34
Description de la fonction RemoveCoachFromHisTeam	35
Description de la fonction GetArbitres	35
Description de la fonction IsAdmin	35
Description de la fonction GetAllTeams	36
Description de la fonction GetTeamById	36
Description de la fonction GetTeamByCoachId	36

Description de la fonction UpdateTeam	37
Description de la fonction GetAllMatchForTeamOnDayAndTime	37
Description de la fonction GetTimes	37
Description de la fonction GetAllTimesOnDay	38
Description de la fonction getTimeById	38
Description de la fonction GetTimeForMatch	38
<b>Tests et protocole de tests</b>	<b>40</b>
Plan de tests	40
Index.php	40
Day.php	40
Match.php	40
Classement.php	41
Equipes.php	41
Staff.php	41
Reset.php	42
MyTeam.php	42
MesMatchs.php	42
NewGame.php	43
addTeamsToGame.php	43
DeleteStaff.php	43
Deletematch.php	43
EditScore.php	43
Rapport de tests	45
Index.php	45
Day.php	45
Match.php	45
Classement.php	46
Equipes.php	46
Staff.php	47
Reset.php	47
MyTeam.php	48
MesMatchs.php	48
NewGame.php	49
addTeamsToGame.php	49
DeleteStaff.php	49
Deletematch.php	49
EditScore.php	49
<b>Conclusion</b>	<b>50</b>
Mon avis sur le TPI	50
Les points positifs	51
Les points à améliorer	51
<b>Liens et références</b>	<b>52</b>
Liens	52
Personnes ressources	52
Documents	52



# INTRODUCTION

Ce document technique fait référence au projet « KidsGames Geneva Score ». Ce projet a été effectué dans le cadre du TPI au centre de formation professionnelle technique. Cette application web a été développée en PHP pour la partie logique et en HTML pour la partie visuelle.

## LES KIDSGAMES

Les KidsGames sont un camp de journée pour les jeunes de 7 à 14 ans. Ils ont lieu pendant les vacances d'été au Parc des Eaux à Genève. Les activités sont très diverses et commencent le matin par une histoire biblique racontée de manière théâtrale (ce qui amuse grands et petits) ainsi que par des chansons. L'après-midi est orienté plus sportif, car c'est à ce moment-là que les enfants jouent aux sports proposés par les KidsGames: Le Tchouckball, le Kinball et la course agile.



## MON PROJET

### RÉSUMÉ

Le « KidsGames Geneva Score » est une application web qui a pour but de faciliter l'enregistrement des scores par les arbitres à la fin de chaque match.

### POURQUOI J'AI CHOISI CE PROJET

J'ai choisi de réaliser ce projet, car je suis devenu responsable des arbitres depuis 2017 et ça a toujours été difficile pour moi de noter tous les scores de tous les matchs durant les 5 minutes de pause que nous avons entre deux matchs. Cette application permet aux arbitres de rester sur leur terrain pendant le changement des équipes tout en enregistrant le score du match qui vient d'avoir lieu. Depuis mon ordinateur, je pourrai ainsi voir tous les scores. Le site permettrait également aux coachs de voir les matchs de leur équipe ainsi que leur position dans le classement.



# CAHIER DES CHARGES

---

## OBJECTIF

---

Ce projet permet aux participants de camps de vacances d'organiser des matches/tournois de sport par équipe

## FONCTIONNALITÉS À RÉALISER

---

Le TPI est une application WEB (php + mysql) et téléphonie mobile (accès au serveur par navigateur) pour permettre aux différents acteurs (organisateurs, arbitres, joueurs) d'organiser les équipes, les rencontres et d'enregistrer les résultats des matches. Le système gère par CRUD les lieux, les personnes, les sports, les équipes, les profils et les scores.

L'administrateur gère les acteurs du site.

## TRAVAIL À RENDRE

---

Pour la fin du TPI, selon calendrier:

### Au secrétariat de l'école

- 1 carnet de bord
- 1 exemplaire papier du mémoire de travail, avec en annexe
  - Documentation technique
  - Mode d'emploi
  - Code source

### Par messagerie aux experts et au maître d'apprentissage

- PDFs du mémoire, de la documentation technique, du mode d'emploi et des sources imprimés du logiciel

### Par messagerie ou autre support électronique au maître d'apprentissage

- les sources (projet logiciel) et la version exécutable

### Pour le jour de la présentation

- Les documents utilisés pour la présentation par messagerie au maître d'apprentissage

### Pour la fin juin

- Résumé du TPI sur une à deux pages, en vue du concours suisse des TPI

## PLANNING

Le travail du TPI à été réalisé en suivant un planning qui est le suivant:

Jour	1	2	3	4	5	6	7	8	9	10
Demi-Journée	1	2	3	4	5	6	7	8	9	10
Analyse, conception de la base.										
Utilisateurs										
Crud Produits										
Gestion des scores et de matches.										
Tests										
Finalisation du site + corrections										
Assemblage final										
Documentation										
Finalisation / Impressions										
Journal										

## ANALYSE FONCTIONNELLE

Dans cette partie de la documentation, vous trouvez toutes les informations de la partie visible du point de vue de l'utilisateur final de l'application. En premier lieu, seront énumérées les fonctions utilisateur (actions GUI). La GUI sera ensuite présentée. Finalement, les mesures de sécurité misent en place seront énumérées.

### DESCRIPTION GLOBALE

Cette application web permet de gérer le planning des matchs de la semaine ainsi que leurs scores. Les coachs pourront facilement visualiser le planning de leur équipe pour savoir quand et où ils doivent se rendre avec leurs joueurs. Les arbitres auront un accès pour facilement voir les matchs qu'ils arbitrent ainsi qu'y mettre un score.

### DESCRIPTION DÉTAILLÉE

Cette application web permet de créer un planning de la semaine (du lundi au samedi) pour les KidsGames. On peut alors visualiser tous les jours et y voir les matchs qui ont lieu. Un utilisateur peut également voir les horaires, les terrains et les équipes des matchs. Les coachs auront un accès plus direct vers le planning de leur équipe (planning accessible par tout le monde). Quant aux administrateurs, ils pourront gérer tout le camp, c'est-à-dire les membres du staff, les matchs, les équipes, etc.

Ci-dessous la liste des actions réalisables par l'utilisateur de l'application d'après son rôle:

**En tant qu'invité**

- Visualiser les jours
- Visualiser les matchs d'un jour
- Visualiser les équipes, le terrain, l'arbitre ainsi que l'heure de début et de fin de chaque match
- Visualiser les équipes avec leur nom et leur coach
- Visualiser le planning de chaque équipe
- Se connecter

## En tant que Coach

- Lien direct vers le planning de son équipe

## En tant que Arbitre

- Visualiser ses matchs
- Enregistrer un score pour ses matchs
- Se déconnecter

## En tant qu'administrateur

- Visualiser les membres du staff
- Ajouter, modifier et supprimer un membre du staff
- Ajouter ou supprimer un match
- Réinitialiser le camp
- Modifier le score d'un match terminé
- Se déconnecter

# DESCRIPTION DE L'INTERFACE

---

## LE TEMPLATE

Le template utilisé pour l'application web s'appelle **Light Wave**. Il est développé par Design Bootstrap et sans droits d'auteur. Il est disponible en six versions différentes sur le site de Graygrids<sup>1</sup>.



*Design Bootstrap*  
Free Bootstrap Templates &  
Snippets For Commercial Use

## PRÉSENTATION DE L'INTERFACE UTILISATEUR

---

<sup>1</sup> <http://demo.graygrids.com/#LightWave>

Le fond d'écran de toutes les pages change d'image toutes les 5 secondes (deux images tournent en boucle). La première image (le carnet de note), est une image qui a servi de bannière lors de l'édition des KidsGames 2016. La deuxième image a été prise lors de l'édition 2017 aux Eaux. L'application web a été



Bannière KidsGames 2016



KidsGames 2017 - Constantin Herrmann

désignée de telle manière à être sobre et moderne. Pour se faire, le « flat style », c'est-à-dire des composants plats et sans ombres se différenciant uniquement grâce à la couleur à l'opacité des composants ont été utilisé. Le site est redimensionnable, il peut ainsi facilement être consulté depuis un ordinateur, une tablette ou un smartphone.

## LA BARRE DE NAVIGATION

Sur toutes les pages de l'application web accessibles à l'utilisateur sera affichée une barre de navigation qui change suivant le rôle de l'utilisateur.

### Invité (Non connecté)



### Administrateur



### Coachs



### Arbitre



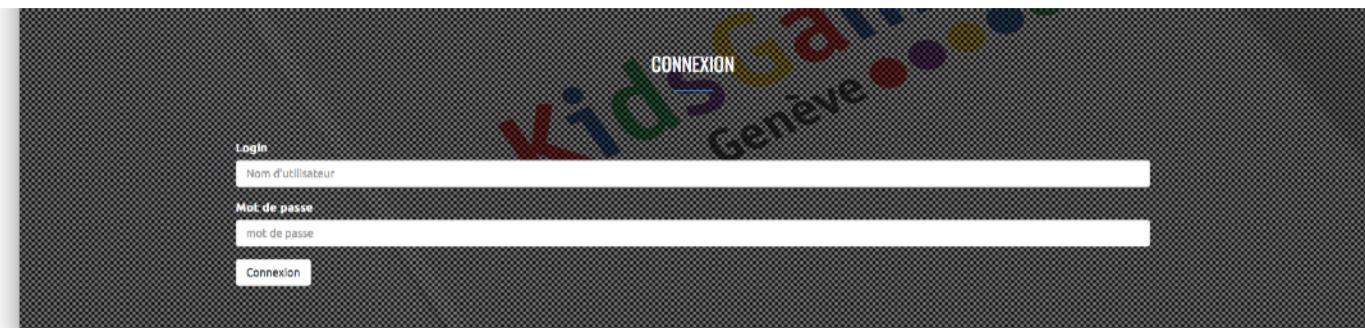
## PAGE D'ACCUEIL

La page d'accueil du site affiche tous les jours du camp (du lundi au samedi) sous forme de petites cartes. Le nom du jour sera affiché en grand (sert également de lien pour arriver sur la page détails du jour), ainsi que le nombre de matchs ayant lieu ce jour-là.



Page index du site

En bas de la page d'index se trouve la section de connexion, seuls les utilisateurs créés par un administrateur peuvent se connecter à l'aide de leur prénom en tant que nom d'utilisateur et le nom de famille comme mot de passe.



Page index - section de connexion

## PAGE CLASSEMENT

La page classement a pour but, comme son nom l'indique, d'afficher les différents classements des trois sports. Tout les utilisateurs de l'application du site peuvent visualiser les classements.

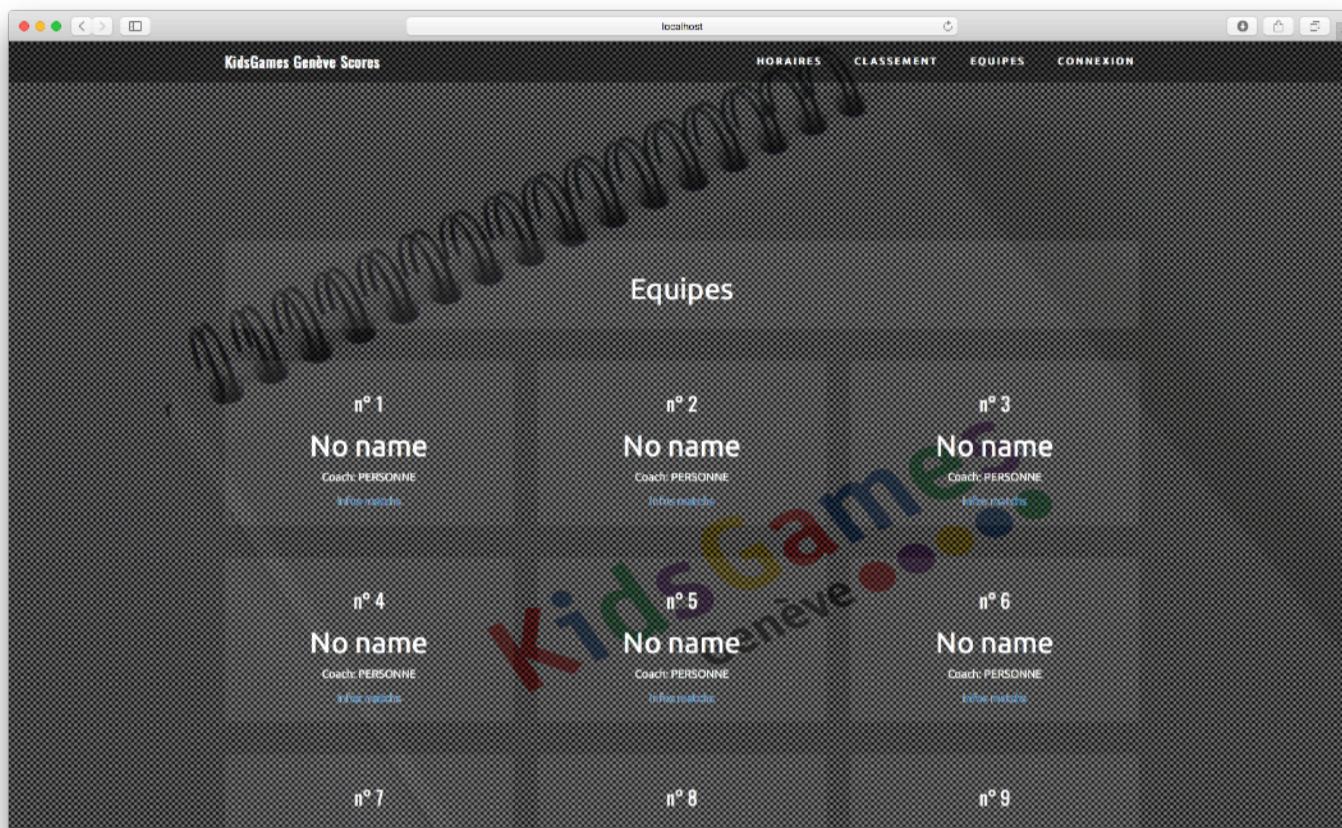
The screenshot displays the 'CLASSEMENT' (Classification) page for 'KINBALL'. The title 'KINBALL' is centered above a table showing 18 entries. The columns are labeled 'Pos.' (Position), 'Nom/num.' (Name/number), 'P.' (Points), 'M.' (Matches), 'R.' (Results), and 'D.' (Decisions). All entries show 'No name (X)' in the 'Nom/num.' column and '0' in all other columns.

Pos.	Nom/num.	P.	M.	R.	D.
1	No name (1)	0	0	0	0
2	No name (17)	0	0	0	0
3	No name (16)	0	0	0	0
4	No name (15)	0	0	0	0
5	No name (14)	0	0	0	0
6	No name (13)	0	0	0	0
7	No name (12)	0	0	0	0
8	No name (11)	0	0	0	0
9	No name (10)	0	0	0	0
10	No name (9)	0	0	0	0
11	No name (8)	0	0	0	0
12	No name (7)	0	0	0	0
13	No name (6)	0	0	0	0
14	No name (5)	0	0	0	0
15	No name (4)	0	0	0	0
16	No name (3)	0	0	0	0
17	No name (2)	0	0	0	0
18	No name (18)	0	0	0	0

La page classement

## PAGE ÉQUIPE

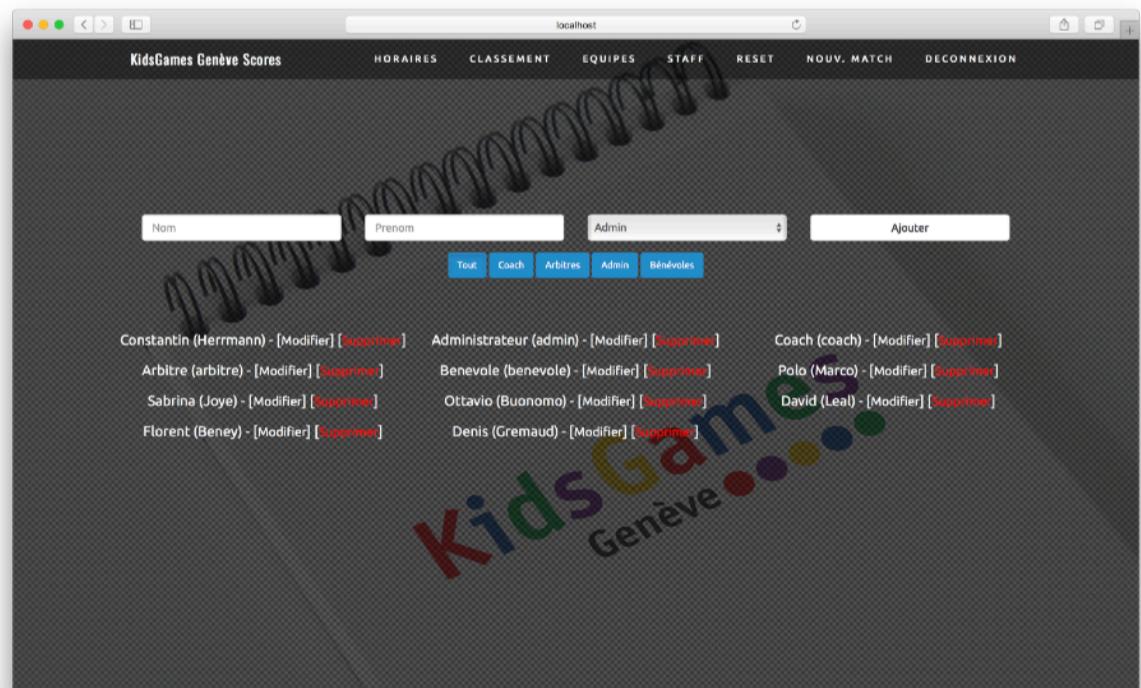
La page équipe affiche toutes les équipes, leur coach si elle en possède un, ainsi qu'un lien redirigeant vers le planning de l'équipe pour la semaine.



Page des équipes

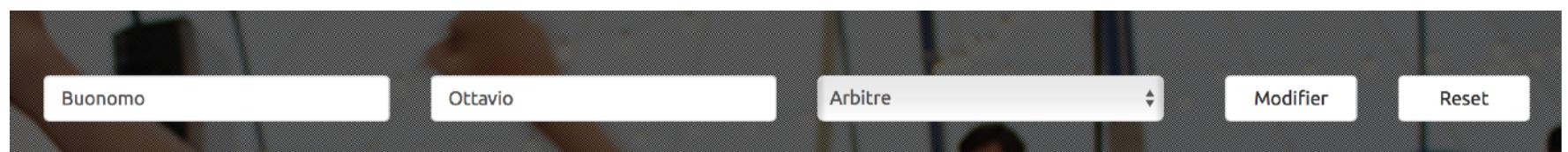
## PAGE STAFF

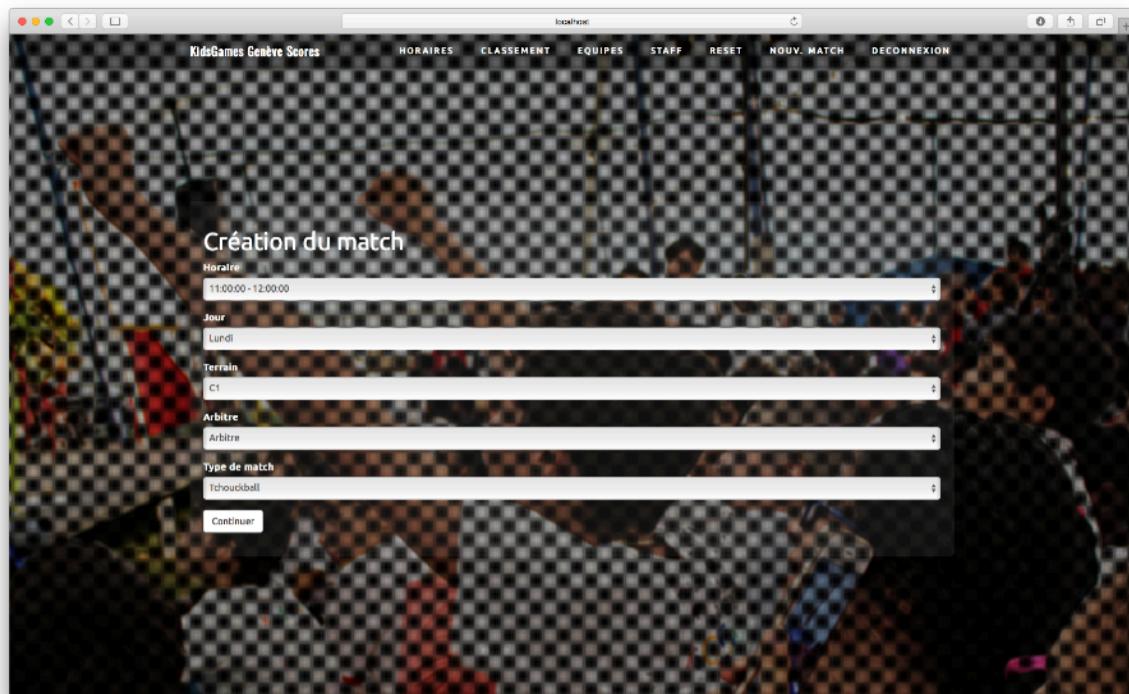
Les administrateurs ont accès à la page staff qui leur permet de visualiser, modifier ou supprimer des membres du staff. Cette page sert à créer des nouveaux comptes pour les nouveaux coachs et nouveaux arbitres. Elle permet également de faire des changements dans le staff déjà existant (par exemple changer le rôle de la personne).



Ils peuvent ajouter ou modifier les membres du staff.

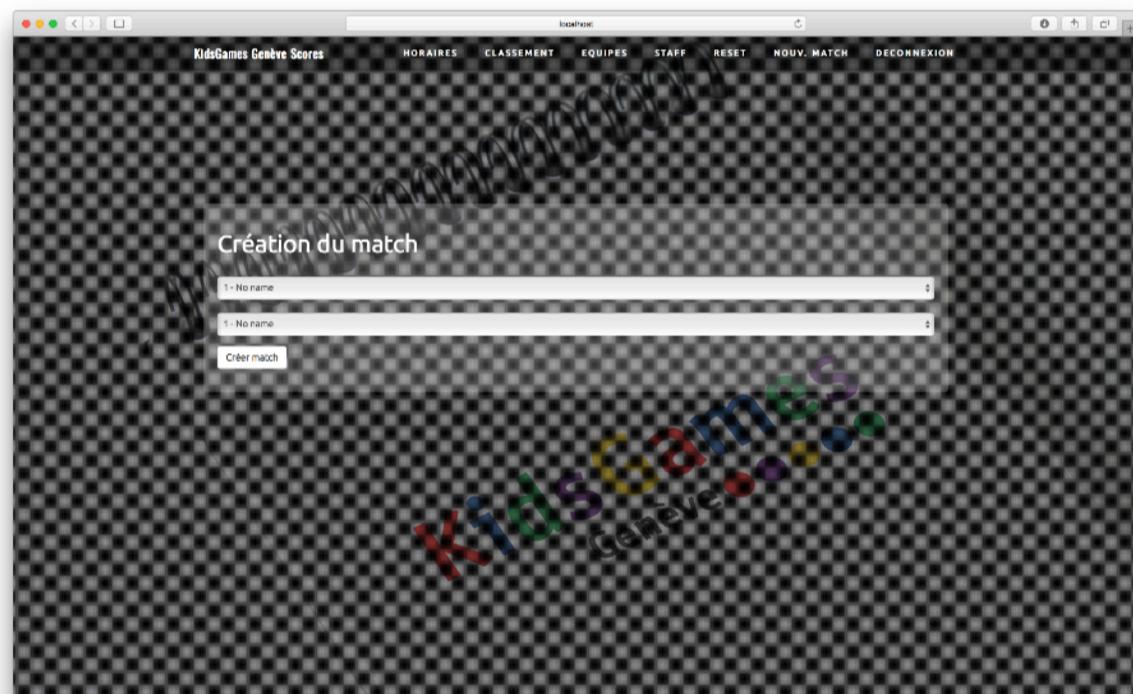
Page du staff





Page création

Une fois toutes les options choisie, appuyer sur le bouton « Continuer » qui redirigera vers la page du choix des équipes.



## PAGE NOUVEAU MATCH

La page nouveau match n'est accessible que par les administrateurs et permet de créer de nouveaux matchs en choisissant toutes les options du matchs qui y correspondent.

Les options à choisir sont les suivantes:

### L'heure

- 11:00:00 - 12:00:00
- 13:20:00 - 14:15:00
- 14:20:00 - 15:15:00
- 15:40:00 - 16:35:00
- 16:40:00 - 17:15:00

### Le jour

- Lundi
- Mardi
- Mercredi
- Jeudi
- Vendredi
- Samedi

### Le terrain

- C1
- C2
- C3
- T1
- T2
- T3
- T4
- T5
- T6
- K1
- K2
- K3
- A1 / A2
- A3 / A4
- A5 / A6

### L'arbitre

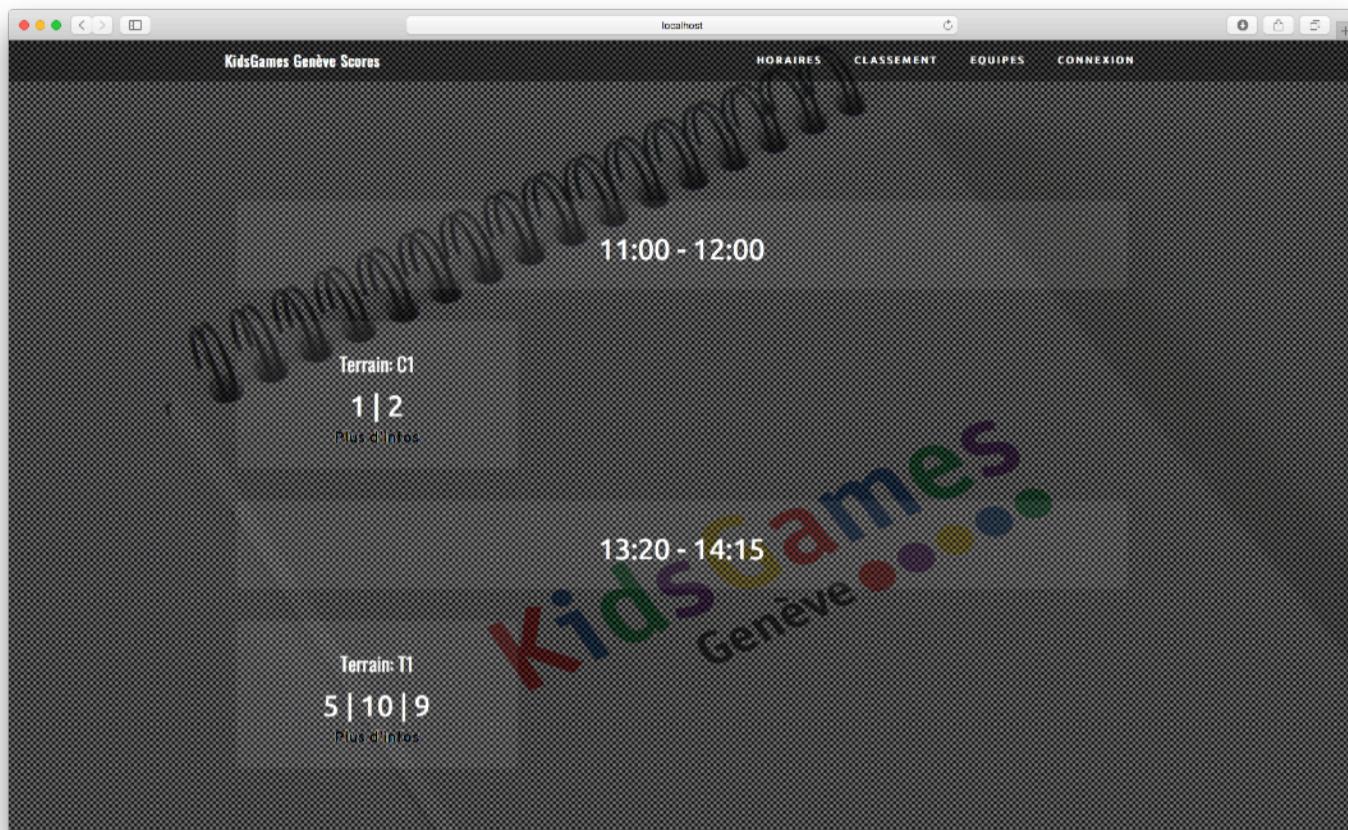
- Arbitre
- David
- Ottavio

### Le type de Sport

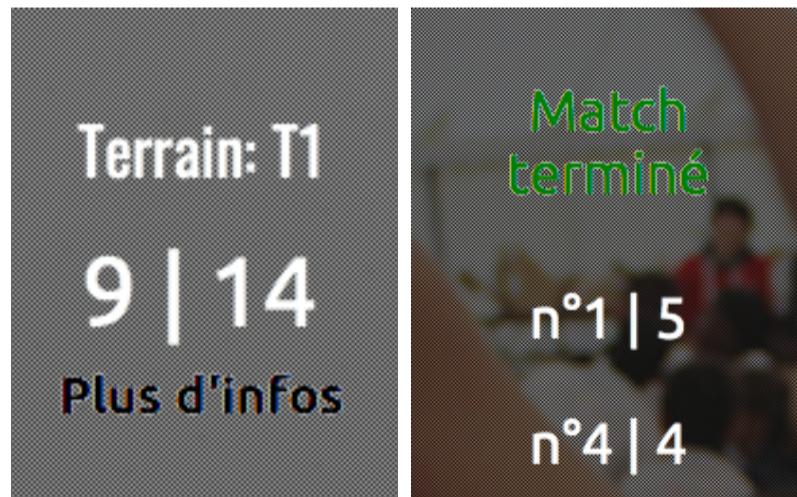
- Tchouckball
- Kinball
- Course-Agile
- Atelier

## PAGE DU JOUR

La page jour affiche tous les matchs du jour sélectionné. La page s'ouvre seulement si le lien contient un ID.



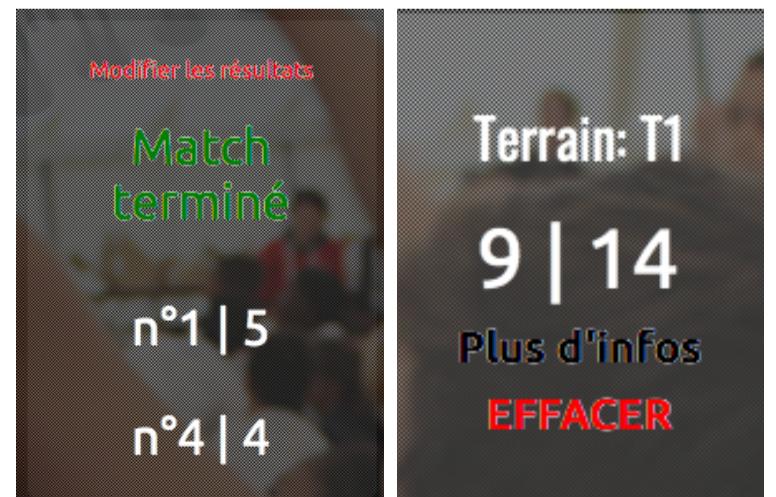
La page affiche tous les matchs sous forme de petites cartes triées par heure de début de début de match. Il y a quatre type de cartes d'affichage possibles.



Les deux premiers types sont ceux qui s'affichent lorsque la page est consultée par un utilisateur non connecté ou un utilisateur qui n'est pas administrateur.

Les deux suivants sont les mêmes que ceux visibles par un utilisateur qui n'est pas administrateur mais par un administrateur. Il possède un lien pour supprimer un match qui n'a pas encore été joué et un lien pour modifier le score d'un match terminé.

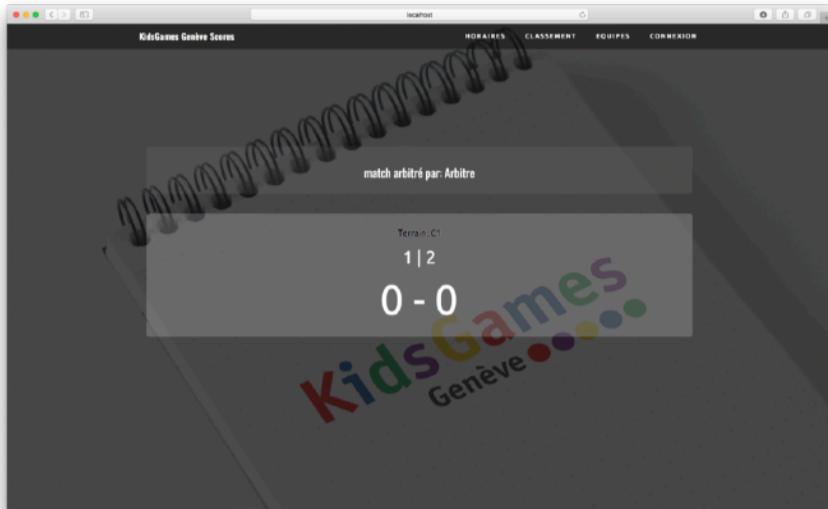
Il n'est pas possible de supprimer un match terminé



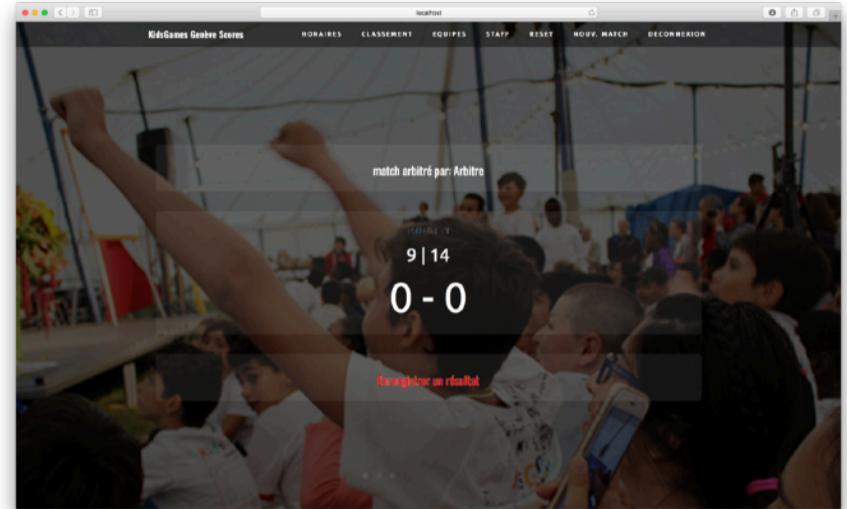
## PAGE MATCH

La page match affiche toutes les informations du match en question. C'est-à-dire que l'utilisateur va trouver le nom de l'arbitre, le nom du terrain, les deux ou trois équipes qui jouent ainsi que le score.

Si l'utilisateur est connecté avec un compte administrateur ou le compte de l'arbitre responsable, il pourra enregistrer un score au match.



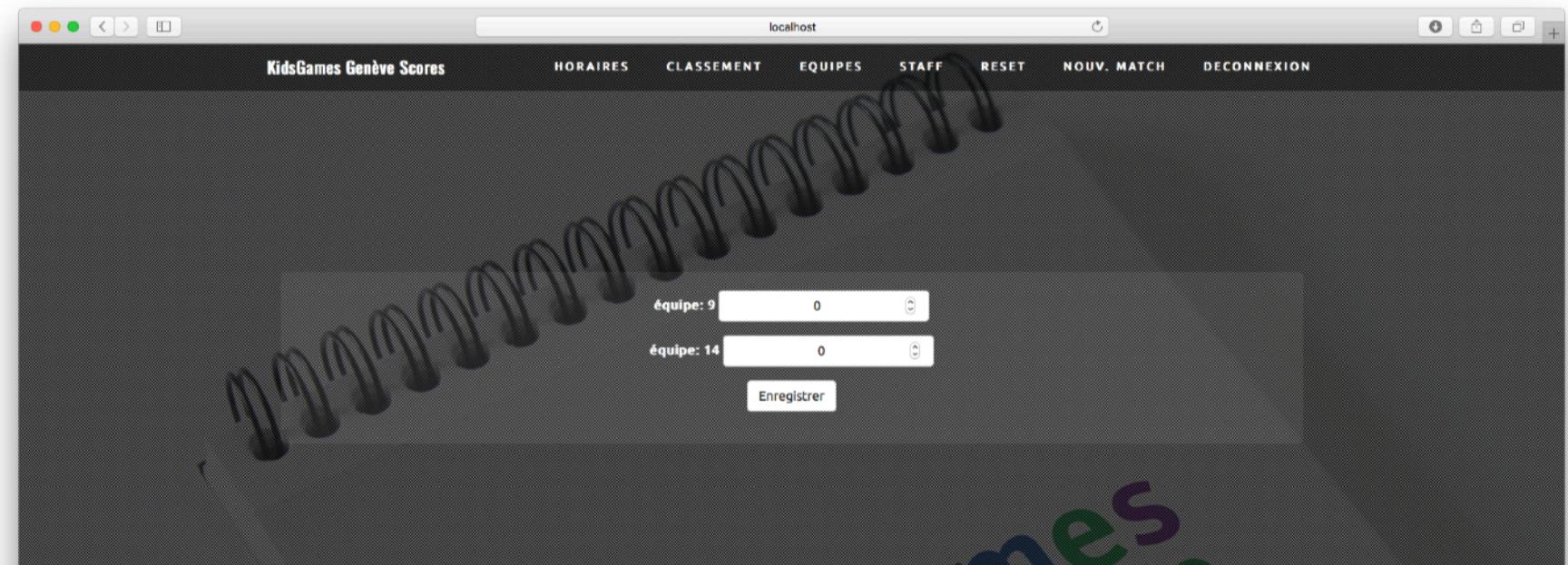
Vue utilisateur standard ou non connecté



Vue administrateur

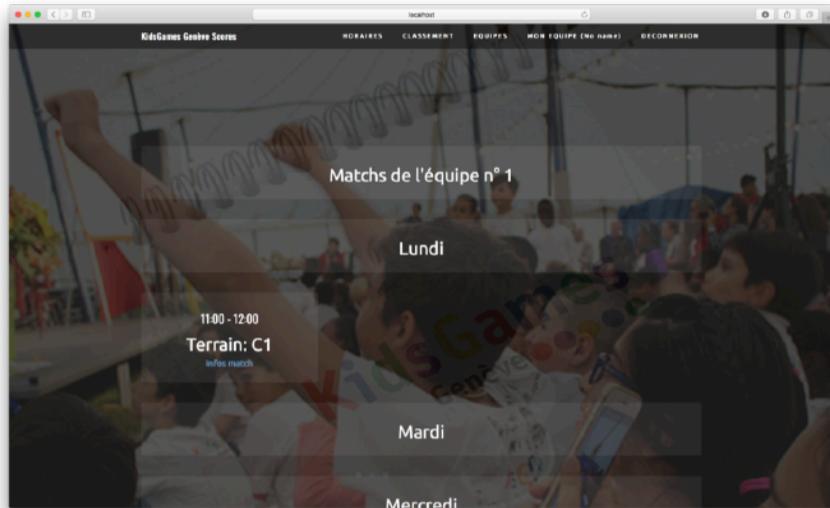
## PAGE MODIFICATION DU SCORE

Cette page est réservée uniquement à l'arbitre responsable du match et aux administrateurs. Elle permet d'enregistrer un score pour les équipes à la fin du match. Un fois le score enregistré, il a le statut « terminé », ce qui bloque l'accès de l'arbitre responsable à la modification du score. Le seul moyen de changer un score par la suite est de passer par un administrateur.

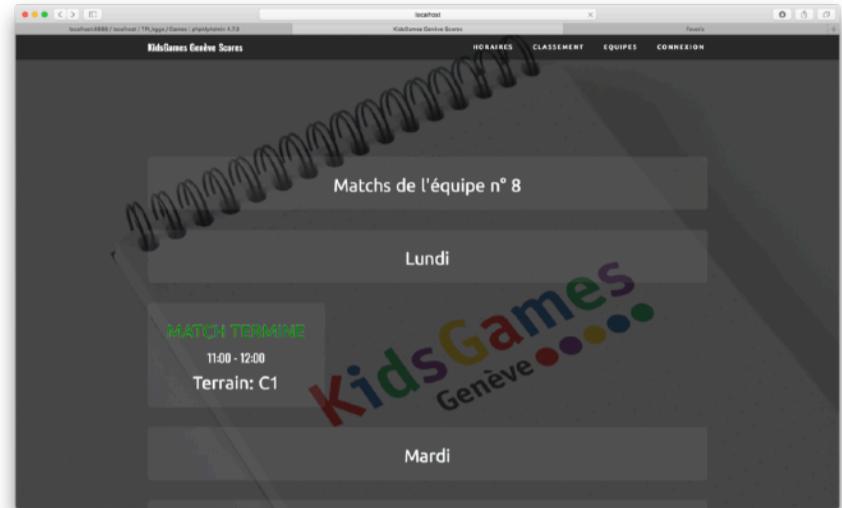


## PAGE DU PLANNING D'UNE ÉQUIPE

La page du planning d'une équipe affiche tout le planning de la semaine pour une équipe. Si le match n'a pas encore été joué, on pourra voir un lien « plus d'infos » redirigeant vers la page du match. Au contraire, si le match est terminé, il y aura juste un message affichant « match terminé ».



Match pas encore joué ou terminé



Match terminé

## PAGE DE MES MATCHS

Cette page est réservée aux arbitres. Elle affiche tous les matchs que l'utilisateur doit arbitrer.

Lorsque les matchs ne sont pas terminés, l'arbitre peut facilement enregistrer un résultat en cliquant sur le lien « Enregistrer un score » où il sera redirigé vers la page editScore.php avec l'id du match.

Si le match est déjà terminé, alors un message s'affichera indiquant qu'il est terminé.



# ANALYSE ORGANIQUE

L'analyse organique décrit le fonctionnement interne de l'application, le "comment". Il s'agit d'analyser ce qui se passe en interne, du point de vue du développeur. En premier lieu sont énumérées les actions implicites du programme, c'est-à-dire ce que l'application doit savoir faire dans le dos de l'utilisateur afin de répondre à ses actions par un résultat cohérent. L'analyse fonctionnelle comporte aussi un passage en revue de toutes les procédures et fonctions afin d'en présenter les buts et le fonctionnement technique (sans lignes de code).

## ARCHITECTURE

A la racine du site se trouvent toutes les pages du site, c'est-à-dire, toutes les pages où l'utilisateur aura accès. A la racine, il y a également deux dossiers, un dossier assets qui contient tous les fichiers de la présentation et la mise en page du site et un dossier PHP contenant tous les fichiers PHP qui sont inclus dans les vues. On retrouve également un autre dossier dans le répertoire PHP nommé controllers. Ce dossier contient tous les fichiers avec les fonctions. Les fonctions sont triées dans des fichiers séparés d'après leurs requêtes sur la base de données.

## BASE DE DONNÉES

Pour que ce projet puisse fonctionner, une base de donnée est demandée. Dans cette section vous trouverez toutes les informations concernant la base de donnée.

### INFORMATIONS

**Type de serveur:** MySQL

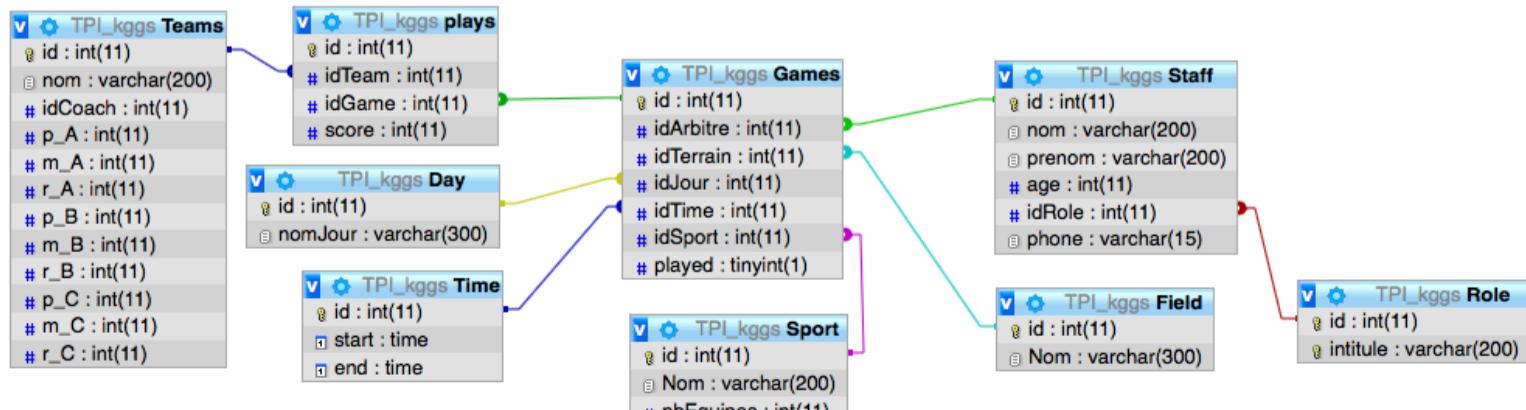
**Version du serveur:** 5.6.35 - MySQL Community Server (GPL)

**Version du protocole:** 10

### ACCÈS

Pour accéder à la base de données depuis les fichiers php, l'application WEB passe par l'objet PDO en lui fournissant les données d'accès requis.

### MCD



La base de données contient neuf tables:

1. **Teams** contient le numéro, le nom, l'id du coach ainsi que toutes les informations des classements de chaque équipe.
2. **Games** contient les informations relatives à un match. L'id de l'arbitre responsable, l'id du terrain, l'id du jour et l'id de l'heure. Il contient également l'id du sport et un champ booléen nommé played qui indique si le match n'a pas encore été joué (0) ou si il est terminé (1).
3. **Day** est la table des jours contenant toutes les informations relatives aux jours. Chaque jour contient un id ainsi qu'un nom (Lundi, Mardi, Mercredi, ...).
4. **Time** possède toutes les heures des matchs. Chaque enregistrement possède un id, une heure de début ainsi qu'une heure de fin.
5. **Sport** tient les informations concernant les sports. La table Sport comporte trois champs: un id, le nom du sport et le nombres d'équipes qui jouent dans le match.
6. **Staff** est la table des membres du staff, chaque staff contient un nom, un prénom, un âge (non utilisé), un id d'un rôle et un numéro de téléphone (non utilisé également).
7. **Field** contient toutes les informations des terrains. Chaque terrain est composé d'un id ainsi qu'un nom.
8. **Role** contient les informations des rôles que les membres du staff peuvent avoir. Un rôle contient un id et un intitulé.
9. **Plays** est la table qui fait la liaison entre une équipe et un match. Un enregistrement de plays contient un id d'une équipe ainsi que l'id du match auquel l'équipe participe.

## SÉCURITÉ DES VUES

---

Le site a été sécurisé au niveau HTML mais également au niveau PHP pour éviter toute tentative de triche.

### PAGE NOUVEAU MATCH

La création du match a été sécurisée à plusieurs niveaux

**1<sup>er</sup>:** Tous les champs sont « required »

**2<sup>ème</sup>:** Une vérification est faite au niveau du php. On vérifie si tous les ID entrés ou sélectionnés par l'utilisateur correspondent à un élément existant.

En cas d'une seule erreur, le formulaire est réinitialisé et un message d'erreur s'affiche au centre de l'écran indiquant que les données ont été entrées de manière incorrecte.

### PAGE MATCH

Si un match est terminé et que l'arbitre responsable du match essaie d'accéder à la page par l'url, celui-ci sera alors directement rediriger sur la page index du site. En revanche, si un administrateur tente d'accéder à la page par l'url, il sera rediriger sur la page de modification des scores du match.

## PAGE MODIFICATION DU SCORE

Plusieurs mesures de sécurité ont été prises pour protéger la page de l'édition des scores. Tout d'abord l'accès est limité aux utilisateurs autorisés aux moments clés.

- Uniquement l'arbitre responsable ainsi que les administrateurs ont accès à la page lorsque celui-ci n'a pas encore été joué
- Une fois le match terminé, uniquement les administrateurs ont accès à la page
- Tout utilisateur non autorisé sera redirigé vers la page index

La seconde partie des mesures de sécurité concerne l'enregistrement des scores.

- Les inputs sont protégés par des required au niveau de l'HTML
- Au niveau du PHP il y a deux sécurités:
  - Si une entrée n'est pas un numéro numérique alors la valeur par défaut est mise. La valeur par défaut est 0.
  - Si une entrée est numérique mais inférieur à 0, alors la valeur par défaut est appliquée.

## PAGE PLANNING D'UNE ÉQUIPE

Si l'ID n'est pas spécifié dans l'url ou qu'il ne correspond pas à une équipe du camp, l'utilisateur sera rediriger sur la page index.

## PAGE DE MES MATCHS

La page est uniquement accessible par les arbitres, c'est pourquoi au lancement de la page, le PHP va récupérer dans la session l'ID de l'utilisateur pour vérifier si celui-ci est bien un arbitre avant d'effectuer la recherche des matchs. Dans le cas où ce n'est pas un arbitre, l'utilisateur sera redirigé sur la page index du site.

## SCRIPT DE SUPPRESSION D'UN MEMBRE DU STAFF

La première vérification effectuée est la vérification de la présence d'un id dans l'url. Si celui-ci est présent, alors l'application va vérifier si l'utilisateur est bien connecté et qu'il s'agit d'un administrateur.

Si tout est en ordre, alors le membre du staff est supprimé de la base de données.

Dans le cas contraire, l'utilisateur sera redirigé vers la page de la gestion du staff.

## SCRIPT DE SUPPRESSION D'UN MATCH

La première vérification effectuée est la vérification de la présence d'un id dans l'url. Si celui-ci est présent, alors l'application va vérifier si l'utilisateur est bien connecté et qu'il s'agit d'un administrateur.

Si les deux premiers tests passent, le match va être vérifié pour s'assurer qu'il n'a pas encore été joué. Dans le cas où celui-ci aurait déjà été joué, ou dans un autre cas de problème pouvant arriver aux tests décrits ci-dessus, l'utilisateur sera redirigé sur la page d'index.

## PAGE ADMINISTRATION DES MEMBRES DU STAFF

La page est exclusivement réservée au membre du staff administrateurs. Tout autre membre ou utilisateur non connecté seront redirigé automatiquement sur la page d'index.

## MÉTHODE DE CODAGE

---

### PREPARE STATEMENT

Le projet a été développé avec des « prepare statement » dans les méthodes accédant à la base de données. Cette méthode a été choisie, car elle permet d'initialiser qu'une seule fois la requête souhaitée. Une fois initialisée, celle-ci restera static dans la page.

Exemple de fonction « prepare statement »

```
/**  
 * Récupère le jour d'apres son id  
 *  
 * @param string l'id du jour  
 * @return array un tableau avec les infos du jour  
 *          ['id'] -> l'id du jour  
 *          ['nomJour'] -> le nom du jour  
 */  
function GetDayById($id){  
    static $query = null;  
    if ($query == null) {  
        $req = "SELECT `id`, `nomJour` FROM `Day` WHERE `id` = :id";  
        $query = connecteur()->prepare($req);  
    }  
    try {  
        $query->bindParam(':id', $id, PDO::PARAM_STR);  
        $query->execute();  
        $res = $query->fetchAll(PDO::FETCH_ASSOC);  
    }  
    catch (Exception $e) {  
        error_log($e->getMessage());  
        $res = false;  
    }  
    return $res;  
}
```

### NOMS ET DESCRIPTIONS

Les noms et les descriptions du code ont été faites de manière complète comme M. Alec Beney me l'a conseillé lors de son atelier TPI. C'est pourquoi, mes méthodes comportent des noms d'une certaine longueur par moment qui ont l'avantage qu'en lisant le code principale on comprend ce que fait la fonction sans devoir aller lire le détail.

## EN-TÊTE DES FICHIERS

Chaque fichier de mon application web comporte une en-tête décrivant les points suivants:

- Le nom et prénom de l'auteur
- La date
- Le titre du projet
- Le titre de la page
- La description de la page
- La version

```
*****
AUTEUR : Constantin Herrmann
LIEU : CFPT Informatique Genève
DATE : 14.06.2018

TITRE PROJET: KidsGames Geneva Score

TITRE PAGE : index
DESCRIPTION : Cette page est les page principale du site internet,
              lorsque qu'un utilisateur accède au site, il arrivera sur cette page.

                  La page affiche tout les jours de la semaine sous forme de petites cartes.
                  Elle affiche également la sectionnde connection.

VERSION : 1.0
*****
```

## INTERACTIONS AVEC L'UTILISATEUR

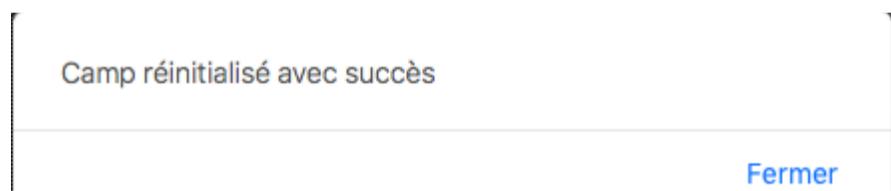
L'application communique à quelques reprises avec l'utilisateur au niveau des erreurs.

### MAUVAIS MOT DE PASSE OU IDENTIFIANT



Ce popup s'affichera lorsque l'utilisateur aura entré un mauvais identifiant et / ou mot de passe sur la page d'accueil dans la section de connexion.

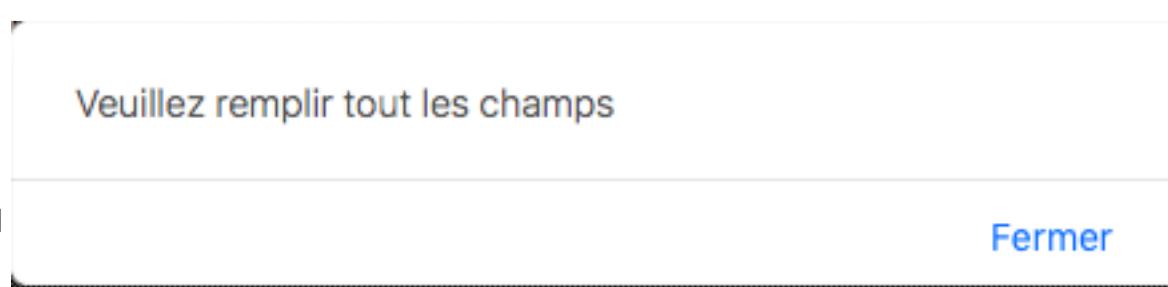
### RÉINITIALISATION DU CAMP



Si le mot de passe super-administrateur est entré correctement, un popup affichera un message indiquant que le camp a été réinitialisé avec succès.



En cas d'erreur, l'application affichera un message indiquant que le mot de passe est éronné.



## MODIFICATION ET AJOUT D'UN MEMBRE DU STAFF

Si, lors de la modification de l'ajout d'un membre du staff, l'utilisateur laisse un champ blanc ou modifie les données dans le code source, il recevra un popup affichant qu'il y une erreur avec les champs.

## CREATION D'UN MATCH

Lors de la création d'un match, si l'utilisateur supprime un champ depuis un code source ou ne rempli pas tout les champs correctement puis clique sur le bouton pour passer à l'étape suivante, il verra apparaître au centre de l'écran le message suivant:



## FONCTIONNALITÉS IMPLICITES

- Connecter un utilisateur et le garder dans la session
- Calculer les points pour une victoire, une égalité ou une défaite
- Récupérer tout les jours
- Récupérer toutes les heures d'une journées
- Mettre à jour un équipe
- Récupérer tout les membres du staff
- Mettre à jour un membre du staff
- Supprimer un membre du staff
- Récupérer tout les matchs
- Créer d'un nouveau match
- Récupérer tout les matchs d'une équipe
- Récupérer tout les matchs d'un arbitre
- Récupérer tout les matchs qui ont lieu à un certain jour
- Récupérer le classement d'un sport

## DESCRIPTIONS DES FONCTIONNALITÉS

### DESCRIPTION DE LA FONCTION LOGIN

La fonction de login sert à vérifier si un utilisateur est correctement authentifier ou non. A partir de la vue, nous allons récupérer l'identifiant de connexion ainsi que le mot de passe entrer par l'utilisateur dans le formulaire de connexion. L'identifiant et le mot de passe seront alors filtrer au niveau du php puis envoyer en paramètre à la fonction de login.

La fonction de login va alors envoyer une requête à la base de données avec comme paramètres l'identifiant et le mot de passe (Dans le cadre de mon application il s'agit du prénom comme identifiant et du nom de famille comme mot de passe). Si la requête retourne un utilisateur, c'est que les identifiants et mot de passe ont été entré correctement. Dans le cas contraire, la fonction retourne false. Elle nécessite deux paramètres:

1. L'identifiant de l'utilisateur
2. Le mot de passe

```
"SELECT `id`, `nom`, `prenom`, `age`, `idRole` FROM `Staff` WHERE `prenom` = :id AND `nom` = :pass"
```

## DESCRIPTION DE LA FONCTION RESETCAMP

La fonction remet à zéro tout le camp en cours, c'est-à-dire que tout les matchs seront supprimés, les classements remis à zéro, le nom de chaque équipe sera remis par défaut (No name) et la liaison entre un coach et son équipe sera également supprimée.

Les membres du staff tant qu'à eux ne seront pas supprimés.

```
UPDATE `Teams` SET `nom` = 'No name', `idCoach` = 0, `p_A` = 0, `m_A` = 0, `r_A` = 0, `p_B` = 0, `m_B` = 0, `r_B` = 0, `p_C` = 0, `m_C` = 0, `r_C` = 0; DELETE FROM `Games`
```

## DESCRIPTION DE LA FONCTION GETCLASSEMENT

Cette fonction à pour but de récupérer un classement. Elle prend comme paramètre la lettre du classement:

- **A** pour le Kinball
- **B** pour le Tchouckball
- **C** pour la course-agile

Elle envoie une requête qui va retourner le classement pour le sport en question trier par les points en ordre descendant. Si deux équipes sont à égalités au niveau des points, elles seront départagées (toujours dans la requête) par la différence de but on ordre descendant également.

Le return est un tableau associatif avec un index numérique qui représente l'ordre du classement (0 -> 1er, 1-> 2ème, 2 -> 3ème, etc.). Chaque index possède ensuite les champs suivants:

- ['id'] —> id de l'équipe
- ['nom'] —> nom de l'équipe
- ['points'] —> les points de l'équipe
- ['marques'] -> score marqué par l'équipe
- ['recus'] —> nombre de score encaisser
- ['difference'] —> la différence des scores

En cas de mauvaise lettre entrée, la fonction retourne false.

```
SELECT `id`, `nom`, `p_classement` as points, `m_classement` as marques, `r_classement` as recus, `m_classement` - `r_classement` as difference FROM `Teams` ORDER BY `p_classement` DESC, `m_classement` - `r_classement` DESC
```

## DESCRIPTION DE LA FONCTION GETDAYS

La fonction récupère tout les jours dans la table ‘Day’ de la base de données. Elle retourne tout les jours dans un tableau associatif indexé. Chaque index du tableau contient:

- ['id'] —> id du jour
- ['nom'] —> nom du jour

```
SELECT `id`, `nomJour` FROM `Day`
```

## DESCRIPTION DE LA FONCTION GETMATCHSONDAYANDTIME

Cette fonction a pour but de récupérer tout les matchs qui ont lieu un certain jour à une certaine heure. Elle prend deux paramètres:

1. L'id du jour
2. L'id de l'heure.

Ces deux paramètres sont ensuite mis dans une requête qui va alors interroger la base de données.

Si il y a des match, alors la fonction retourne un tableau associatif indexé ou chaque index contient un id.

**Exemple:** \$unMatch = \$matchs[1]['id'];

Si aucun match n'as lieu, la valeur renournée est égal à false.

```
SELECT `id` FROM `Games` WHERE `idJour` = :day AND `idTime` = :time
```

## DESCRIPTION DE LA FONCTION GETDAYBYID

La fonction GetDayById à pour but de retourner les informations du jour (en l'occurrence, son nom), d'après un id donné en paramètre.

Si l'id existe dans la table, alors la fonction va retourner un tableau associatif contenant les champs suivants:

- ['id'] —> l'id du jour
- ['nomJour'] —> le nom du jour

Si l'id n'est pas valide ou pas dans la base de données, elle retournera false.

```
SELECT `id`, `nomJour` FROM `Day` WHERE `id` = :id
```

## DESCRIPTION DE LA FONCTION GETFIELDS

Le but de la fonction est de récupérer tout les terrains dans la base de données. Ils seront alors retourné sous forme de tableau associatif indexé. Chaque index possédera un id ainsi que le nom du terrain.

- ['id'] —> l'id du terrain
- ['nom'] —> le nom du terrain

```
SELECT `id`, `Nom` FROM `Field`
```

## DESCRIPTION DE LA FONCTION GETFIELDBYID

Cette fonction retourne les informations d'un terrain d'après son id. La fonction envoie l'id reçu en paramètre dans la requête envoyée à la base de données.

Si un terrain à été trouvé à l'id donné, alors la fonction va retourner un tableau associatif contant deux champs:

- ['id'] —> l'id du terrain
- ['nom'] —> le nom du terrain

Dans le cas où l'id du terrains ne serait pas valide ou inexistant dans la table, la fonction retournera false.

```
SELECT `id`, `Nom` FROM `Field` WHERE `id` = :id
```

## DESCRIPTION DE LA FONCTION CREATENEWGAME

CreateNewGames à pour but de créer un nouveau match. Cette fonction nécessite six paramètres:

1. L'id de l'heure
2. L'id du jour
3. L'id du terrain
4. L'id de l'arbitre
5. Un tableau avec les id des équipes (teams[index] => 1)
6. L'id du sport

La première étape est de créer le match dans la base de données. La fonction va alors récupérer l'id du dernier élément insérer dans la base de données.

```
INSERT INTO `Games`(`idArbitre`, `idTerrain`, `idJour`, `idTime`, `idSport`) VALUES (:arbitre,:terrain,:jour,:heure,:sport)
```

La deuxième étape est la création des référence pour chaque équipe dans la table plays. Pour ce faire, la fonction va parcourir le tableau des id des équipes (5ème paramètre) pour les ajouter dans la table 'plays' avec l'id du match.

```
INSERT INTO `plays`(`idTeam`, `idGame`) VALUES (:team, :game)
```

## DESCRIPTION DE LA FONCTION GETALLMATCHSINFOS

Cette à fonction à pour but de récolter toutes les informations pour un match souhaité. La fonction nécessite un paramètre:

## 1. L'id du match

L'id sera alors envoyé dans les requêtes pour récupérer les informations du match ainsi que les équipes qui jouent le match et le score obtenu.

Si l'id est valide, alors la fonction va aller chercher les informations du match pour les stocker dans un tableau associatif.

```
SELECT `Games`.`id` as id, `Games`.`played` as played, `Field`.`nom` as terrain,
`Staff`.`prenom` as arbitre FROM `Games` INNER JOIN `Field` ON `idTerrain` = `Field`.`id`
INNER JOIN `Staff` ON `idArbitre` = `Staff`.`id` WHERE `Games`.`id` = :id LIMIT 1
```

La fonction va ensuite faire une autre requête pour récupérer les équipes qui jouent au match et leur score pour les stocker dans un deuxième tableau associatif.

```
SELECT `Teams`.`id` as numero, `Teams`.`nom` as nom, `score` FROM `Teams`, `plays` WHERE
`Teams`.`id` IN (SELECT `idTeam` FROM `plays` WHERE `idGame` = :id) AND `plays`.`idGame` =
:id AND `Teams`.`id` = `plays`.`idTeam`
```

Les deux tableaux vont alors être mis dans un tableau associatif global. C'est-à-dire que le tableau des informations se trouvera dans la case 'infos' et les équipes dans la case 'teams' du tableau global.

La fonction va retourner le tableau global rempli si l'id est valide, sinon la fonction retournera false.

## DESCRIPTION DE LA FONCTION ISARBITREORADMIN

Cette fonction vérifie si l'utilisateur est soit l'arbitre du match, soit un administrateur. Les paramètres requis pour la fonction sont:

1. L'id du staff
2. L'id du match

La fonction va alors envoyer une requête vérifiant si l'id du staff correspond à l'id de l'arbitre responsable du match en question.

```
SELECT `Staff`.`prenom` FROM `Staff` WHERE `Staff`.`id` = (SELECT `Games`.`idArbitre` FROM
`Games` WHERE `Games`.`id` = :idGame LIMIT 1) AND `Staff`.`id` = :idStaff
```

Si c'est le cas, alors la fonction return true.

Si l'id du staff ne correspond pas à celui attendu, alors la fonction va retourner le résultat de la fonction « IsAdmin »(retourne un booléen) en envoyant l'id du staff reçu comme paramètre de l'autre fonction.

## DESCRIPTION DE LA FONCTION COUNTGAMESONDAY

Cette fonction retourne le compte de tous les matchs qui ont lieu durant une journée. Pour ce faire, la fonction requiert un paramètre:

1. L'id du jour

Si l'id du jour est valable, alors la fonction va retourner un tableau associatif avec le compte des matchs:

- ['matchs'] —> le nombre de matchs

Dans le cas contraire, la fonction retournera false.

```
SELECT COUNT(*) as matchs FROM `Games` WHERE `idJour` = :idDay
```

## DESCRIPTION DE LA FONCTION DELETGAME

Cette fonction supprime un match avec l'id indiqué en paramètre. C'est le seul paramètre requis pour supprimer un match.

Grâce aux relations de la base, supprimer un match supprimera aussi les liaisons de la table plays.

```
DELETE FROM `Games` WHERE `id` = :id
```

## DESCRIPTION DE LA FONCTION GETMATCHSFORARBITREDAYANDTIME

Cette fonction récupère tout les matchs pour un arbitre pour un jour à une heure.

Les paramètres requis pour la fonctions:

1. L'id du staff (l'arbitre)
2. L'id du jour
3. L'id de l'heure

La fonction va alors envoyer une requête à la base pour récupérer les matchs que l'arbitre doit arbitre au jour et à l'heure indiquée.

```
SELECT `Games`.`id` as id, `Field`.`Nom` as terrain, `Games`.`played` as played FROM `Games`  
INNER JOIN `Field` ON `Field`.`id` = `Games`.`idTerrain` WHERE `Games`.`idArbitre`  
= :idStaff AND `Games`.`idJour` = :idDay AND `Games`.`idTime` = :idTime LIMIT 1
```

Si le résultat de la requête contient quelque chose, la fonction retournera les matchs, sinon la fonction retournera false.

## DESCRIPTION DE LA FONCTION UPDATESCORE

La fonction de updatescore se charge de mettre à jour les scores ainsi que le classement. Cette fonction est la plus chargée et la plus complexe de l'application web.

Cette fonction requiert deux paramètres:

1. Un tableau associatif avec les infos des équipes ainsi que les scores

```
$teams[index]['infos']['numero'];
$teams[index]['infos']['nom'];
$teams[index]['score'];
(L'index représente une équipe)
```

2. L'id du match

Dans un premier temps, la fonction va commencer par récupérer toutes les infos du match dans la base de données. Elle va ensuite récupérer toutes les données des équipes et de leur score actuelles du match.

Ensuite, elle va actualiser le match pour le régler en tant que « terminé ». Pour ce faire, la fonction va envoyer une requête pour changer le champ ‘played’ de 0 à 1 grâce à la fonction « SetMatchToDo ». Puis elle va à nouveau récupérer toutes les informations du match.

Dans un troisième temps, la fonction va parcourir toutes les équipes reçus. Pour chaque équipe, le programme va effectuer les action suivantes:

- Récupérer l'id de l'équipe et le stocké dans une variable provisoire
- Récupérer le score de l'équipe et le stocké dans une variable provisoire
- Mettre à jour le score de l'équipe dans la table de liaison ‘plays’

```
UPDATE `plays` SET `score` = :score WHERE `idTeam` = :idTeam AND `idGame` = :idGame
```

Par la suite, le programme va récupérer les nouveaux résultats du match afin d'avoir les nouveaux résultats en plus des anciens résultats.

Avant de repartir dans une boucle qui parcours toutes les équipes. La fonction va récupérer le type de classement pour en créer une variable.

Le programme parcours à nouveau toutes les équipes. Cette fois-ci pour chaque équipe, il effectue d'autres actions:

- Recherche de l'ancien score obtenu dans la liste des anciens scores
- Recherche du nouveau score dans la liste des nouveaux scores
- Récupère les données du classement principal de l'équipe (fonction « GetMainClassement »)
- Compare le tableau actuel avec les anciens et nouveaux scores
- Met à jour le classement (fonction « UpdateMainTable ») principale avec les résultats des calculs

## DESCRIPTION DE LA FONCTION GETMAINCLASSEMENT

Le but de la fonction est de retourner le classement principale souhaité pour une équipe. Deux paramètres sont requis:

1. L'id de l'équipe
2. La lettre du classement (**A** = Kinball, **B** = Tchouckball, **C** = Course-agile)

Si la requête envoyée aboutis à un résultat, alors la fonction retournera un tableau associatif:

- [‘id’] —> L'id de l'équipe
- [‘p’] —> Le nombre de points pour l'équipe
- [‘m’] —> Les buts marqués
- [‘r’] —>Les buts reçus

Sinon, la fonction retournera false.

```
SELECT `id`, `p_`.{$classment}` as p, `m_`.{$classment}` as m, `r_`.{$classment}` as r FROM `Teams` WHERE `id` = :idTeam
```

## DESCRIPTION DE LA FONCTION GETMATCHRESULTS

Cette fonction va récupérer les résultats du match en renvoyant un tableau associatif en incluant les points qui ont été calculé par les fonctions « MakeTheResults » et « MakeThePoints ». Elle prend deux paramètres:

1. L'id du match
2. Un tableau associatif avec toutes les infos du match en question

Cette fonction à pour tâche de récupérer tout les scores et les équipes pour les envoyé dans la fonction « MakeTheResult » qui ce chargera de faire la suite.

```
SELECT `idTeam` as id, `score` FROM `plays` WHERE `idGame` = :idGame ORDER BY `score` DESC
```

Les scores sont trier de manière descendant, de tel manière à ce que l'équipe avec le plus de points marqué soit en premier position du tableau.

La fonction retourne un tableau associatif indexé. Chaque index possèdera les champs suivants:

- ['id'] —> L'id de l'équipe
- ['points'] —> Le nombre de points pour l'équipe
- ['m'] —> Les buts marqués
- ['r'] —> Les buts reçus
- ['d'] —> La différence entre marqués et reçus

## DESCRIPTION DE LA FONCTION MAKETHERESULTS

Cette fonction est appelée depuis la fonction « GetMatchResults ». Elle prend deux paramètres:

1. Un tableau associatif avec les infos et les scores des équipes

```
$teams[index]['infos']['numero'];
$teams[index]['infos']['nom'];
$teams[index]['score'];
```

2. Un tableau associatif avec les infos du match

La fonction va alors calculer pour chaque équipes les résultats obtenus, c'est à dire les points marqués, reçus et la différence pour les stocker dans un tableau associatif nommé 'results'.

Elle va ensuite appeler la fonction « MakeThePoints » en envoyant le tableau avec les information du match ainsi que le tableau 'results'.

La fonction retourne un tableau associatif indexé. Chaque index possèdera les champs suivants:

- ['id'] —> L'id de l'équipe

- ['points'] —> Le nombre de points pour l'équipe
- ['m'] —> Les buts marqués
- ['r'] —> Les buts reçus
- ['d'] —> La différence entre marqués et reçus

### DESCRIPTION DE LA FONCTION MAKETHETPOINTS

La fonction décrite ici est la dernière fonction nécessaire pour calculer les résultats d'un match. Elle est appelée par la fonction « MakeTheResult » et nécessite deux paramètres:

1. Un tableau associatif des résultats des équipes classé par résultats obtenus
2. Le tableau associatif avec les informations du match

Le programme va alors commencer par vérifier si il y a deux ou trois équipes qui jouent. Il va ensuite passer à la phase de comparaison des scores pour attribuer les points à chaque équipe.

Pour rappel:

- **Victoire** = 3 points
- **Égalité** = 1 point
- **Défaite** = 0 points

La fonction retourne un tableau associatif indexé. Chaque index possèdera les champs suivants:

- ['id'] —> L'id de l'équipe
- ['points'] —> Le nombre de points pour l'équipe
- ['m'] —> Les buts marqués
- ['r'] —> Les buts reçus
- ['d'] —> La différence entre marqués et reçus

### DESCRIPTION DE LA FONCTION UPDATERMAINTABLE

Cette fonction est appelée juste après le calcul des résultats depuis la fonction « UpdateScores ». Elle permet de mettre à jour le classement principal. Elle requiert cinq paramètres:

1. L'id de l'équipe
2. Les points à ajouter
3. Les points marqués
4. Les points reçus
5. Le label du classement dans lequel il faut actualiser les résultats

```
UPDATE `Teams` SET `p_classement` = $points_to_add, `m_classement` = $points_m, `r_classement` = $points_r WHERE `id` = $idTeam
```

Comme aucune action de l'utilisateur n'est possible, il n'y a pas besoin d'effectuer un bindParam depuis le PDO.

## DESCRIPTION DE LA FONCTION GETTYPEOFCLASSEMENT

L'objectif de cette fonction est de fournir l'id du sport pour un match. Un seul paramètre requis:

1. L'id du match

Si l'id est valable, la fonction retournera un tableau associatif:

- ['sport'] —> L'id du sport

Sinon, elle retournera false.

## DESCRIPTION DE LA FONCTION SETGAMETODONE

Cette fonction à comme unique but de changer le champ played d'un match de 0 à 1. En d'autres termes, elle change le statut du match de « Pas joué » à « terminé ». Pour son fonctionnement elle nécessite un paramètre:

1. L'id du match

```
UPDATE `Games` SET `played` = 1 WHERE `id` = :id
```

## DESCRIPTION DE LA FONCTION GETSCOREFORTEAMINMATCH

Cette fonction retourne le score d'une équipe pour un match. Deux paramètres sont requis:

1. L'id de l'équipe
2. L'id du match

Si les données sont valables, alors la fonction retournera un tableau associatif:

- ['score'] —> Le score de l'équipe

Sinon, elle retournera false.

```
SELECT `score` FROM `plays` WHERE `idTeam` = :idTeam AND `idGame` = :idGame
```

## DESCRIPTION DE LA FONCTION GETROLEBYID

Le but de cette fonction est de retourner l'intitulé d'un rôle d'après son id. Elle nécessite un paramètre:

1. L'id du rôle

Si les données sont valables, alors la fonction retournera un tableau associatif:

- ['intitule'] —> Le nom du rôle

Sinon, elle retournera false.

```
SELECT `intitule` FROM `Role` WHERE `id` = :id LIMIT 1
```

**DESCRIPTION DE LA FONCTION GETALLROLES**

Cette fonction récupère tout les rôles dans la base de données.

Si la table contient des données, alors la fonction retournera un tableau associatif indexé. Chaque index possédera les champs suivants:

- ['id'] —> L'id du rôle
- ['intitule'] —> Le nom du rôle

Sinon, elle retournera false.

```
SELECT `id`, `intitule` FROM `Role`
```

**DESCRIPTION DE LA FONCTION GETSPORTTYPES**

La fonction à pour bit de récupérer tout les types de sports dans la base de données.

Si la table contient des données, alors la fonction retournera un tableau associatif indexé. Chaque index possédera les champs suivants:

- ['id'] —> L'id du sport
- ['Nom'] —> Le nom du sport
- ['nbEquipes'] —> Le nombres d'équipes nécessaires pour un match

Sinon, elle retournera false.

```
SELECT `id`, `Nom`, `nbEquipes` FROM `Sport`
```

**DESCRIPTION DE LA FONCTION GETSPORTBYID**

Cette fonction récupères les infos d'un sport d'après son id. Elle nécessite un paramètre:

1. L'id du sport

Si les données sont valables, alors la fonction retournera un tableau associatif:

- ['id'] —> L'id du sport
- ['Nom'] —> Le nom du sport
- ['nbEquipes'] —> Le nombres d'équipes nécessaires pour un match

Sinon, elle retournera false.

```
SELECT `id`, `Nom`, `nbEquipes` FROM `Sport` WHERE `id` = :id
```

**DESCRIPTION DE LA FONCTION GETIDSBYPLAYERCOUNT**

Cette fonction récupère les sports de la base de données d'après le nombre d'équipes nécessaires pour que le match ai lieu. Elle requiert un paramètre:

1. Le nombre de joueurs

Si les données sont valables, alors la fonction retournera un tableau associatif indexé. Chaque index contiendra les champs suivants:

- ['id'] —> L'id du sport
- ['Nom'] —> Le nom du sport
- ['nbEquipes'] —> Le nombres d'équipes nécessaires pour un match

Sinon, elle retournera false.

```
SELECT `id`, `Nom`, `nbEquipes` FROM `Sport` WHERE `nbEquipes` = :amount
```

### **DESCRIPTION DE LA FONCTION GETSTAFFBYID**

Cette fonction récupère les informations d'un membre du staff d'après son id. Un seul paramètre est requis:

1. L'id du membre du staff

Si les données sont valables, alors la fonction retournera un tableau associatif indexé. Chaque index contiendra les champs suivants:

- ['id'] —> l'id du membre
- ['nom'] —> le nom
- ['prenom'] —> le prénom
- ['age'] —> l'âge
- ['idRole'] —> l'id de son Rôle
- ['phone'] —> le numéro de téléphone

Sinon, elle retournera false.

```
SELECT `id`, `nom`, `prenom`, `age`, `idRole`, `phone` FROM `Staff` WHERE `id` = :id
```

### **DESCRIPTION DE LA FONCTION GETALLSTAFF**

Cette fonction récupère les informations de tout les membres du staff dans la base de données. Aucun paramètre.

Si la table contient des données, alors la fonction retournera un tableau associatif indexé. Chaque index possédera les champs suivants:

- ['id'] —> l'id du membre
- ['nom'] —> le nom
- ['prenom'] —> le prénom
- ['age'] —> l'âge
- ['idRole'] —> l'id de son Rôle

- ['phone'] —> le numéro de téléphone

Sinon, elle retournera false.

```
SELECT `id`, `nom`, `prenom`, `age`, `idRole`, `phone` FROM `Staff`
```

## DESCRIPTION DE LA FONCTION ADDSTAFF

Cette fonctionnalité permet d'ajouter un membre au staff. Elle nécessite trois paramètres:

1. Le nom
2. Le prénom
3. L'id du rôle

(Toutes les données auront été vérifiées avant l'envoie dans cette fonction.)

```
INSERT INTO `Staff`(`nom`, `prenom`, `idRole`) VALUES (:nom, :prenom, :idRole)
```

## DESCRIPTION DE LA FONCTION UPDTESTAFF

Cette fonctionnalité permet de mettre à jour un membre au staff. Elle nécessite quatre paramètres:

1. L'id de l'utilisateur à modifier
2. Le nom
3. Le prénom
4. L'id du rôle

(Toutes les données auront été vérifiées avant l'envoie dans cette fonction.)

```
INSERT INTO `Staff`(`nom`, `prenom`, `idRole`) VALUES (:nom, :prenom, :idRole)
```

## DESCRIPTION DE LA FONCTION DELETESTAFF

DeleteStaff permet de supprimer un membre du staff. Elle requiert un paramètre:

1. Id du staff

Avant de supprimer le membre du staff, le programme va regarder si le membre est un coach et si il possède une équipe. Si c'est le cas, la liaison entre le coach et son équipe sera supprimée grâce à la fonction de « RemoveCoachFromHisTeam ».

Une fois la liaison supprimée (si il en existait une), le membre du staff sera supprimé.

```
DELETE FROM `Staff` WHERE `id` = :id
```

## DESCRIPTION DE LA FONCTION GETALLCOACHSWITHOUTTEAM

Cette fonction récupère tous les coachs de la base de données qui n'ont pas de liaison avec une équipe.

Si la table contient des données, alors la fonction retournera un tableau associatif indexé. Chaque index possédera les champs suivants:

- ['id'] —> l'id du membre
- ['nom'] —> le nom
- ['prenom'] —> le prénom

Sinon, elle retournera false.

```
SELECT `id`, `nom`, `prenom` FROM `Staff` WHERE `idRole` = 2 AND `id` NOT IN (SELECT idCoach FROM TEAMS WHERE id != 0) ORDER BY `prenom`
```

### **DESCRIPTION DE LA FONCTION REMOVECOACHFROMHISTEAM**

Cette fonction permet de supprimer la liaison entre un coach et son équipe. Elle remet à -1 le champ 'idCoach' de la table 'Teams'. Elle requiert un paramètre:

1. L'id du coach

La requête sera alors envoyée à la base données.

```
UPDATE `Teams` SET `idCoach` = -1 WHERE `idCoach` = :idCoach
```

### **DESCRIPTION DE LA FONCTION GETARBITRES**

Cette fonction récupère tout les membres du staff qui sont arbitres. Donc où l'id du rôle est égal à 3. Cette fonction ne nécessite aucun paramètre.

Si la table contient des données, alors la fonction retournera un tableau associatif indexé. Chaque index possédera les champs suivants:

- ['id'] —> l'id du membre
- ['nom'] —> le nom
- ['prenom'] —> le prénom

Sinon, elle retournera false.

```
SELECT `id`, `nom`, `prenom` FROM `Staff` WHERE `idRole` = 3 ORDER BY `prenom`
```

### **DESCRIPTION DE LA FONCTION ISADMIN**

Cette fonction sert à vérifier si un utilisateur est administrateur ou non. Elle requiert un paramètre:

1. L'id du de l'utilisateur

Si les données sont valides, alors la fonction retournera true. Sinon, false.

```
SELECT `Staff`.`prenom` FROM `Staff` WHERE `Staff`.`idRole` = (SELECT `Role`.`id` FROM `Role` WHERE `Role`.`intitule` = 'Admin') AND `Staff`.`id` = :id
```

**DESCRIPTION DE LA FONCTION GETALLTEAMS**

Cette fonction récupère les infos de toutes les équipes dans la base de données.

Si la table contient des données, alors la fonction retournera un tableau associatif indexé. Chaque index possédera les champs suivants:

- ['id'] —> l'id de l'équipe
- ['nom'] —> le nom de l'équipe
- ['idCoach'] —> l'id du coach correspondant à l'équipe
- ['total'] —> le total de points
- ['totalDiff'] —> le total de la différence

Sinon, elle retournera false.

```
SELECT `id`, `nom`, `idCoach`, `p_A` + `p_B` + `p_C` as total, `m_A` - `r_A` + `m_B` - `r_B` + `m_C` - `r_C` as totalDiff FROM `Teams` ORDER BY `id` ASC
```

**DESCRIPTION DE LA FONCTION GETTEAMBYID**

Le but de cette fonction est de récupérer toutes les infos d'une équipe d'après un id. Elle nécessite un paramètre:

1. L'id de l'équipe

Si les données sont valables, alors la fonction retournera un tableau associatif:

- ['id'] —> l'id de l'équipe
- ['nom'] —> le nom de l'équipe
- ['idCoach'] —> l'id du coach correspondant à l'équipe

Sinon, elle retournera false.

```
SELECT `id`, `nom`, `idCoach` FROM `Teams` WHERE `id` = :id LIMIT 1
```

**DESCRIPTION DE LA FONCTION GETTEAMBYCOACHID**

Le but de cette fonction est de récupérer toutes les infos d'une équipe d'après l'id du coach. Elle nécessite un paramètre:

1. L'id du coach

Si les données sont valables, alors la fonction retournera un tableau associatif:

- ['id'] —> l'id de l'équipe
- ['nom'] —> le nom de l'équipe

Sinon, elle retournera false.

```
SELECT `id`, `nom` FROM `Teams` WHERE `idCoach` = :id LIMIT 1
```

## DESCRIPTION DE LA FONCTION UPDATETEAM

Cette fonction met à jour les données d'une équipe. Elle nécessite trois paramètres:

1. L'id de l'équipe
2. Le nom de l'équipe
3. L'id du coach

```
UPDATE `Teams` SET `nom` = :nom, `idCoach` = :coach WHERE `id` = :id
```

## DESCRIPTION DE LA FONCTION GETALLMATCHFORTEAMONDAYANDTIME

Cette fonction récupère les matchs pour une équipe pour un jour et une heure. Elle nécessite trois paramètres:

1. L'id de l'équipe
2. L'id du jour
3. L'id du fuseau horaire

Si les données sont valables, alors la fonction retournera un tableau associatif:

- ['id'] —> l'id du match (game)
- ['arbitre'] —> le nom de l'arbitre qui arbitre le match
- ['terrain'] —> le nom du terrain
- ['idJour'] —> l'id du Jour
- ['idTime'] —> l'id du fuseau horaire
- ['Nom'] —> le nom du sport que l'équipe joue pour le match
- ['played'] —> 1 si la match a été joué, 0 si non

Sinon, elle retournera false.

```
SELECT `Games`.`id`, `Staff`.`nom` as arbitre, `Field`.`Nom` as terrain, `idJour`, `idTime`, `Sport`.`Nom`, `played` FROM `Games` INNER JOIN `Staff` ON `staff`.`id` = `Games`.`idArbitre` INNER JOIN `Field` ON `Field`.`id` = `Games`.`idTerrain` INNER JOIN `Sport` ON `Sport`.`id` = `Games`.`idSport` WHERE `Games`.`id` IN (SELECT `plays`.`idGame` FROM `plays` WHERE `plays`.`idTeam` = :idTeam) AND `Games`.`idJour` = :idDay AND `Games`.`idTime` = :idTime
```

## DESCRIPTION DE LA FONCTION GETTIMES

Cette fonction récupère toutes les heures dans la base de données. Elle ne requiert aucun paramètre.

Si la table contient des données, alors la fonction retournera un tableau associatif indexé. Chaque index possédera les champs suivants:

- ['id'] —> l'id du fuseaux
- ['start'] —> l'heure de début
- ['end'] —> l'heure de fin

Sinon, elle retournera false.

```
SELECT `id`, `start`, `end` FROM `Time`
```

## DESCRIPTION DE LA FONCTION GETALLTIMESONDAY

Le but de cette fonction est de récupérer toutes les heures pour un jour. Elle nécessite un paramètre:

1. L'id du jour

Si les données entrées sont valables, alors la fonction retournera un tableau associatif indexé. Chaque index possédera les champs suivants:

- ['idTime'] —> l'id de l'heure

Sinon, elle retournera false.

```
SELECT `idTime` FROM `Games` WHERE `idJour` = :idJour GROUP BY `idTime`
```

## DESCRIPTION DE LA FONCTION GETTIMEBYID

Cette fonction récupère les informations d'un jour d'après un id. Elle requiert un paramètre:

1. L'id de l'heure

Si les données entrées sont valables, alors la fonction retournera un tableau associatif:

- ['id'] -> l'id
- ['start'] -> l'heure de début
- ['end'] -> l'heure de fin

Sinon, elle retournera false.

```
SELECT `id`, `start`, `end` FROM `Time` WHERE `id` = :id
```

## DESCRIPTION DE LA FONCTION GETTIMEFORMATCH

Cette fonction récupère les informations d'un jour d'après un id pour un match. Elle requiert un paramètre:

1. L'id du match

Si les données entrées sont valables, alors la fonction retournera un tableau associatif:

- ['debut'] —> l'heure du début
- ['fin'] —> l'heure de fin

Sinon, elle retournera false.

```
SELECT `Time`.%start` as debut, `Time`.%end` as fin FROM `Time` WHERE `Time`.%id` IN  
(SELECT `Games`.%idTime` FROM `Games` WHERE `Games`.%id` = :idGame)
```



Moment de pause entre deux matchs - KidsGames 2017 | Eaux

# TESTS ET PROTOCOLE DE TESTS

## PLAN DE TESTS

Chaque test ci-dessous doit être réaliser su la page indiquée en haut de chaque tableau

### INDEX.PHP

N°	Scénario (enchainement d'actions)	Résultat attendu
1.01	- clic sur sur le nom d'un jour	<ul style="list-style-type: none"> <li>- Redirection sur la bonne page</li> <li>- Récupération des données du bon jour</li> <li>- Affichage du message « aucun match » si le jour ne contient pas de match, sinon affiche tout les matchs</li> </ul>
1.02	- Connexion avec un bon mot de passe	<ul style="list-style-type: none"> <li>- Mise à jour de la barre de navigation en fonction du rôle</li> <li>- La section « connexion » est masquée de la page index</li> </ul>
1.03	- Connexion avec un mauvais mot de passe ou un compte non existant	<ul style="list-style-type: none"> <li>- Popup qui préviens que le nom ou l'utilisateur est erroné</li> <li>- Message en rouge au dessus de la section « connexion » de la page index</li> <li>- Redirection sur la page index ainsi que la section « connexion »</li> </ul>

### DAY.PHP

N°	Scénario (enchainement d'actions)	Résultat attendu
2.01	- Clic sur « plus d'infos » d'un match	- redirection sur la page du match
2.02	- Si il n'y a pas de matchs pour le jour	- Affichage d'un message indiquant qu'il n'y a pas de match
2.03	- Ouverture de la page sans spécifié l'id du jour dans l'url	- redirection sur la page index

### MATCH.PHP

N°	Scénario (enchainement d'actions)	Résultat attendu
3.01	- Ouverture de la page (sans connexion)	<ul style="list-style-type: none"> <li>- Récupère correctement toutes les données du match : L'arbitre, le terrain, le numéro des équipes, le score</li> </ul>
3.02	- Ouverture de la page avec un compte administrateur connecté	<ul style="list-style-type: none"> <li>- Récupère correctement toutes les données du match : L'arbitre, le terrain, le numéro des équipes, le score</li> <li>- Affichage d'un lien « Enregistrer un score » redirigeant sur la page editScore.php</li> </ul>
3.03	- Ouverture de la page avec un compte arbitre (Pas celui de l'arbitre responsable du match)	<ul style="list-style-type: none"> <li>- Affichage identique à celui d'une personne non-connectée (n° 3.01)</li> </ul>
3.04	- Ouverture de la page en étant connecté avec le compte de l'arbitre responsable du match	<ul style="list-style-type: none"> <li>- Affichage identique a celui d'un administrateur (n° 3.02)</li> </ul>

N°	Scénario (enchainement d'actions)	Résultat attendu
3.05	- Ouverture de la page sans connexion et le match terminé	- Redirection sur la page index.php
3.06	- Ouverture de la page connecté avec un compte arbitre (y compris le responsable du match) et le match terminé	- Redirection sur la page index.php
3.07	- Ouverture de la page avec un compte administrateur et le match terminé	- Redirection sur la page editScore.php du match

## CLASSEMENT.PHP

N°	Scénario (enchainement d'actions)	Résultat attendu
4.01	- Ouverture de la page (en étant connecté OU non)	- Récupère le contenu des trois classements trier par ordre décroissant des points

## EQUIPES.PHP

N°	Scénario (enchainement d'actions)	Résultat attendu
5.01	- Ouverture de la page (sans être connecté OU connecté avec un compte coach OU connecté avec un compte arbitre)	- Récupère toutes les données de la base de données - Affiche correctement le numéro de l'équipe, le nom de l'équipe, son coach, un lien redirigeant vers le planning du match (myTeam.php) - Si l'équipe ne possède pas de coach, alors un message est affichée laissant passer le message
5.02	- Ouverture de la page en étant connecté avec un compte administrateur	- Mêmes résultats attendus que l'ouverte sans connexion (n° 5.01) - Affichage d'un lien au dessus de l'équipe redirigeant vers une page de modification d'équipe

## STAFF.PHP

N°	Scénario (enchainement d'actions)	Résultat attendu
6.01	- Ouverture de la page (sans être connecté OU connecté avec un compte coach OU connecté avec un compte arbitre)	- Redirection sur la page index.php
6.02	- Ouverture de la page en étant connecté avec un compte administrateur	- Récupère correctement toutes les infos dans la base de données - Affiche tous les membres du staff - Affiche un formulaire pour ajouter et modifier le staff
6.03	- Clic sur le bouton « modifier » à côté d'un membre du staff	- Son nom apparaît dans la case « nom » du formulaire de modification - Son prénom apparaît dans la case « prenom » du formulaire de modification - Son rôle est sélectionné dans le sélect du rôle du formulaire de modification - Le texte du bouton affiche « modifier » - Un autre bouton est affiché servant à reset le formulaire

N°	Scénario (enchainement d'actions)	Résultat attendu
<b>6.04</b>	- Clic sur le bouton « reset » du formulaire lors de la modification d'un membre du staff	- Annule les changements - Réinitialise le formulaire
<b>6.05</b>	- Laisser tout les inputs du formulaire vide - Clic sur le bouton « ajouter » du formulaire	- Envoie un message d'erreur à l'utilisateur - Réinitialise le formulaire

## RESET.PHP

N°	Scénario (enchainement d'actions)	Résultat attendu
<b>7.01</b>	- Ouverture de la page (sans être connecté <b>OU</b> connecté avec un compte coach <b>OU</b> connecté avec un compte arbitre)	- Redirection sur la page index.php
<b>7.02</b>	- Ouverture de la page en étant connecté avec un compte administrateur	- Affichage d'un formulaire demandant un mot de passe
<b>7.03</b>	- Taper un faux mot de passe - Clic sur « Valider Réinitialisation »	- Popup avec un message indiquant que le mot de passe est erroné - Réinitialisation du champ pour entrer le mot de passe
<b>7.04</b>	- Taper le bon mot de passe (KidsGames) - Clic sur « Valider Réinitialisation »	- Popup avec un message indiquant que le camp à bien été réinitialiser - Réinitialisation du champ pour entrer le mot de passe

## MYTEAM.PHP

N°	Scénario (enchainement d'actions)	Résultat attendu
<b>8.01</b>	- Ouverture de la page sans id spécifié dans l'url (myTeam.php?id=...)	- Redirection sur la page index.php
<b>8.02</b>	- Ouverture de la page avec la page avec un id spécifié dans l'url mais erronné	- Redirection sur la page index.php
<b>8.03</b>	- Ouverture de la page avec la page avec un id spécifié dans l'url et correct	- Récupère toutes les infos dans la base de données - Affiche tout les jours - Affiche les matchs sous formes de petites cartes dans le jour correspondant - Affiche un titre avec le numéro de l'équipe

## MESMATCHS.PHP

N°	Scénario (enchainement d'actions)	Résultat attendu
<b>9.01</b>	- Ouverture de la page (sans être connecté <b>OU</b> connecté avec un compte coach <b>OU</b> connecté avec un compte administrateur)	- Redirection sur la page index.php
<b>9.02</b>	- Ouverture de la page en étant connecté avec un compte arbitre	- Récupère toutes les données dans la base de données - Affiche tout les jours - Affiche tout les matchs  - Si les matchs sont terminés, alors « Match terminé » et les résultats sont affichés - Si les matchs ne sont pas terminés, alors un lien pour enregistrer les résultats apparaît et le terrain ainsi que l'heure du match

## NEWGAME.PHP

N°	Scénario (enchainement d'actions)	Résultat attendu
10.01	- Ouverture de la page (sans être connecté <b>OU</b> connecté avec un compte coach <b>OU</b> connecté avec un compte arbitre)	- Redirection sur la page index.php
10.02	- Remplir à moitié le formulaire - Appuyer sur le bouton « suivant »	- Popup avec un message affichant qu'il y a une erreur - Réinitialisation du formulaire
10.03	- Changement des valeurs depuis le code source - Appuyer sur le bouton « suivant »	- Popup avec un message affichant qu'il y a une erreur - Réinitialisation du formulaire

## ADDTTEAMSTOGAME.PHP

N°	Scénario (enchainement d'actions)	Résultat attendu
11.01	- Ouverture de la page (sans être connecté <b>OU</b> connecté avec un compte coach <b>OU</b> connecté avec un compte arbitre)	- Redirection sur la page index.php
11.02	- Accès direct à la page par l'url	- Redirection sur la page index.php

## DELETESTAFF.PHP

N°	Scénario (enchainement d'actions)	Résultat attendu
12.01	- Ouverture de la page sans spécifié l'id dans l'url	- Redirection sur la page index.php
12.02	- Ouverture de la page (sans être connecté <b>OU</b> connecté avec un compte coach <b>OU</b> connecté avec un compte arbitre)	- Redirection sur la page index.php

## DELETEMATCH.PHP

N°	Scénario (enchainement d'actions)	Résultat attendu
13.01	- Ouverture de la page sans spécifié l'id dans l'url	- Redirection sur la page index.php
13.02	- Ouverture de la page (sans être connecté <b>OU</b> connecté avec un compte coach <b>OU</b> connecté avec un compte arbitre)	- Redirection sur la page index.php
13.03	- Accéder à la page avec un id dans l'url d'un match qui n'a pas encore été joué	- Ignorer la suppression - Redirection sur la page index.php
13.04	- Accéder à la page avec toutes les données valides	- Suppression ok - Redirection sur la page index.php

## EDITSCORE.PHP

N°	Scénario (enchainement d'actions)	Résultat attendu
14.01	- Ouverture de la page (sans être connecté <b>OU</b> connecté avec un compte coach <b>OU</b> connecté avec un compte arbitre)	- Redirection sur la page index.php
14.02	- Ouverture de la page sans id précisé dans l'url	- Redirection sur la page index.php

N°	Scénario (enchainement d'actions)	Résultat attendu
14.03	- Ouverture de la page avec l'id d'un match terminé dans l'url et connecté avec un compte non administrateur <b>ou</b> un sans connexion	- Redirection sur la page index.php
14.04	- Ouverture de la page avec l'id d'un match terminé dans l'url et connecté avec un compte administrateur	- Affiche le formulaire pour changer les scores
14.05	- Changer le code source pour autorisé les scores négatifs dans le formulaire - Entrer un score négatif - Clic sur « enregistrer »	- Met la valeur par défaut (0)
14.06	- Changer le code source pour autorisé la saisie de texte dans le formulaire - Entre une chaîne de caractère - Clic sur « enregistrer »	- Met la valeur par défaut (0)
14.07	- Remplir le formulaire correctement - Clic sur « enregistrer »	- Enregistre correctement les données dans la base de données - Redirection sur la page index.php



Franz en train de jouer la batterie pendant les chansons du matin  
KidsGames 2017 | Evaux

## RAPPORT DE TESTS

Voici le rapport des test décrits ci-dessus.

### INDEX.PHP

N°	Résultat	Commentaire	Résultat obtenu
1.01	OK	Aucun	<ul style="list-style-type: none"> <li>- L'utilisateur est redirigé sur la page du jour</li> <li>- La page affiche tout les matchs du jour triés par heure</li> </ul>
1.02	OK	Aucun	<ul style="list-style-type: none"> <li>- La barre de navigation se met à jour en fonction du rôle de l'utilisateur</li> <li>- Sur la page index, la section connexion est masquée</li> <li>- Le bouton de connexion n'est plus disponible dans la barre de navigation et a été remplacé par un bouton « déconnexion »</li> </ul>
1.03	OK	Aucun	<ul style="list-style-type: none"> <li>- Un popup apparaît au milieu de l'écran avec le message suivant « identifiant ou mot de passe erroné »</li> <li>- Un message dans une boîte rouge est visible au dessus de la section de connexion</li> <li>- L'utilisateur est immédiatement redirigé vers la page d'index dans la section connexion</li> <li>- Le formulaire a été réinitialisé</li> </ul>

### DAY.PHP

N°	Résultat	Commentaire	Résultat obtenu
2.01	OK	Aucun	<ul style="list-style-type: none"> <li>- L'utilisateur est correctement redirigé sur la page du match</li> </ul>
2.02	OK	Aucun	<ul style="list-style-type: none"> <li>- Message en haut de la page indiquant qu'il n'y a pas de matchs qu'y ont lieu</li> </ul>
2.03	OK	Aucun	<ul style="list-style-type: none"> <li>- L'utilisateur est correctement redirigé sur la page index</li> </ul>

### MATCH.PHP

N°	Résultat	Commentaire	Résultat obtenu
3.01	OK	Aucun	<ul style="list-style-type: none"> <li>- Toutes les données du match ont été récupérées</li> <li>- Toute les informations du match sont affichées correctement</li> </ul>

N°	Résultat	Commentaire	Résultat obtenu
3.02	OK	Aucun	<ul style="list-style-type: none"> <li>- Toutes les données du match ont été récupérées</li> <li>- Toute les informations du match sont affichées correctement</li> <li>- Un lien est affiché en bas de la page « Enregistrer un score »</li> </ul>
3.03	OK	Aucun	<ul style="list-style-type: none"> <li>- Toutes les données du match ont été récupérées</li> <li>- Toute les informations du match sont affichées correctement</li> </ul>
3.04	OK	Aucun	<ul style="list-style-type: none"> <li>- Toutes les données du match ont été récupérées</li> <li>- Toute les informations du match sont affichées correctement</li> <li>- Un lien est affiché en bas de la page « Enregistrer un score »</li> </ul>
3.05	OK	Aucun	<ul style="list-style-type: none"> <li>- L'utilisateur est correctement rediriger sur la page index</li> </ul>
3.06	OK	Aucun	<ul style="list-style-type: none"> <li>- L'utilisateur est correctement rediriger sur la page index</li> </ul>
3.07	OK	Aucun	<ul style="list-style-type: none"> <li>- L'utilisateur est correctement rediriger sur la page d'édition des scores du match</li> </ul>

## CLASSEMENT.PHP

N°	Résultat	Commentaire	Résultat obtenu
4.01	OK	Aucun	<ul style="list-style-type: none"> <li>- Tout les classements s'affichent correctement dans des tableau distinctifs</li> </ul>

## EQUIPES.PHP

N°	Résultat	Commentaire	Résultat obtenu
5.01	OK	Aucun	<ul style="list-style-type: none"> <li>- Les équipes sont toutes correctement affichées sous forme de petite cartes</li> <li>- Chaque carte contient le numéro de l'équipe, le nom, le nom du coach</li> <li>- Un lien est également affiché dans les cartes : « plus d'infos »</li> <li>- Les équipes qui n'ont pas de coach affichent « Aucun coach »</li> </ul>

N°	Résultat	Commentaire	Résultat obtenu
5.02	OK	Aucun	<ul style="list-style-type: none"> <li>- Les équipes sont toutes correctement affichées sous forme de petite cartes</li> <li>- Chaque carte contient le numéro de l'équipe, le nom, le nom du coach</li> <li>- Un lien est également affiché dans les cartes : « plus d'infos »</li> <li>- Les équipes qui n'ont pas de coach affichent « Aucun coach »</li> <li>- Affichage d'un lien au dessus de la carte « modifier l'équipe »</li> </ul>

## STAFF.PHP

N°	Résultat	Commentaire	Résultat obtenu
6.01	OK	Aucun	<ul style="list-style-type: none"> <li>- L'utilisateur est correctement rediriger sur la page index</li> </ul>
6.02	OK	Aucun	<ul style="list-style-type: none"> <li>- Tous les membres du staff sont affichés correctement</li> <li>- Affiche les bouton modifier et supprimer à coté de chaque membre</li> <li>- Le formulaire d'ajout et de modification est bien affiché</li> </ul>
6.03	OK	Aucun	<ul style="list-style-type: none"> <li>- Toutes les information du membre du staff sont correctement affichées dans les champs correspondants dans le formulaire</li> <li>- Le rôle sélectionner dans le sélect correspond au rôle du membre</li> <li>- Deux bouton son affichés: Modifier, Reset</li> </ul>
6.04	OK	Aucun	<ul style="list-style-type: none"> <li>- Les changements n'ont pas été pris en compte</li> <li>- Le formulaire à été remis à zéro</li> </ul>
6.05	OK	Aucun	<ul style="list-style-type: none"> <li>- Un popup avec le message suivant est affiché au centre de l'écran: « Veuillez remplir tout les champs correctement »</li> <li>- Le formulaire est réinitialisé</li> </ul>

## RESET.PHP

N°	Résultat	Commentaire	Résultat obtenu
7.01	OK	Aucun	<ul style="list-style-type: none"> <li>- L'utilisateur est correctement rediriger sur la page index</li> </ul>

N°	Résultat	Commentaire	Résultat obtenu
7.02	OK	Aucun	<ul style="list-style-type: none"> <li>- Un formulaire demandant le mot de passe super administrateur est affiché ainsi qu'un bouton « réinitialiser »</li> </ul>
7.03	OK	Aucun	<ul style="list-style-type: none"> <li>- Un popup avec le message suivant est affiché au centre de l'écran: « Mot de passe incorrect»</li> <li>- Le formulaire est réinitialisé</li> </ul>
7.04	OK	Aucun	<ul style="list-style-type: none"> <li>- Un popup avec le message suivant est affiché au centre de l'écran: « Le camp à bien été réinitialiser»</li> <li>- Le formulaire est réinitialisé</li> </ul>

## MYTEAM.PHP

N°	Résultat	Commentaire	Résultat obtenu
8.01	OK	Aucun	<ul style="list-style-type: none"> <li>- L'utilisateur est correctement rediriger sur la page index</li> </ul>
8.02	OK	Aucun	<ul style="list-style-type: none"> <li>- L'utilisateur est correctement rediriger sur la page index</li> </ul>
8.03	OK	Aucun	<ul style="list-style-type: none"> <li>- Une entête est affichée en haut de la page</li> <li>- Affiche tout les jours de la semaine</li> <li>- Affiche tout les matchs sous forme de petites cartes</li> <li>- Affiche les infos du match comme l'heure et le terrain aise qu'un lien permettant l'accès au match</li> <li>- Les matchs terminés s'affichent avec le résultat du match</li> </ul>

## MESMATCHS.PHP

N°	Résultat	Commentaire	Résultat obtenu
9.01	OK	Aucun	<ul style="list-style-type: none"> <li>- L'utilisateur est correctement rediriger sur la page index</li> </ul>
9.02	OK	Aucun	<ul style="list-style-type: none"> <li>- Les jours s'affichent correctement</li> <li>- Les matchs sont affichés d'après les jours</li> <li>- Les matchs terminés affichent le message « Match terminé »</li> <li>- Les match non terminés affichent l'heure, le terrain et un bouton « Enregistrer un résultat »</li> </ul>

## NEWGAME.PHP

N°	Résultat	Commentaire	Résultat obtenu
10.01	OK	Aucun	<ul style="list-style-type: none"> <li>- L'utilisateur est correctement rediriger sur la page index</li> </ul>
10.02	OK	Aucun	<ul style="list-style-type: none"> <li>- Un popup avec le message suivant est affiché au centre de l'écran: « Veuillez remplir tout les champs correctement»</li> <li>- Le formulaire est réinitialisé</li> </ul>
10.03	OK	Aucun	<ul style="list-style-type: none"> <li>- Un popup avec le message suivant est affiché au centre de l'écran: « Veuillez remplir tout les champs correctement»</li> <li>- Le formulaire est réinitialisé</li> </ul>

## ADDTTEAMSTOGAME.PHP

N°	Résultat	Commentaire	Résultat obtenu
11.01	OK	Aucun	<ul style="list-style-type: none"> <li>- L'utilisateur est correctement rediriger sur la page index</li> </ul>
11.02	OK	Aucun	<ul style="list-style-type: none"> <li>- L'utilisateur est correctement rediriger sur la page index</li> </ul>

## DELETTESTAFF.PHP

N°	Résultat	Commentaire	Résultat obtenu
12.01	OK	Aucun	<ul style="list-style-type: none"> <li>- L'utilisateur est correctement rediriger sur la page index</li> </ul>
12.02	OK	Aucun	<ul style="list-style-type: none"> <li>- L'utilisateur est correctement rediriger sur la page index</li> </ul>

## DELETEMATCH.PHP

N°	Résultat	Commentaire	Résultat obtenu
13.01	OK	Aucun	<ul style="list-style-type: none"> <li>- L'utilisateur est correctement rediriger sur la page index</li> </ul>
13.02	OK	Aucun	<ul style="list-style-type: none"> <li>- L'utilisateur est correctement rediriger sur la page index</li> </ul>
13.03	OK	Aucun	<ul style="list-style-type: none"> <li>- Ignore la suppression</li> <li>- L'utilisateur est correctement rediriger sur la page index</li> </ul>
13.04	OK	Aucun	<ul style="list-style-type: none"> <li>- Le match à été supprimer</li> <li>- L'utilisateur est correctement rediriger sur la page index</li> </ul>

## EDITSCORE.PHP

N°	Résultat	Commentaire	Résultat obtenu
14.01	OK	Aucun	<ul style="list-style-type: none"> <li>- L'utilisateur est correctement rediriger sur la page index</li> </ul>

N°	Résultat	Commentaire	Résultat obtenu
14.02	OK	Aucun	- L'utilisateur est correctement rediriger sur la page index
14.03	OK	Aucun	- L'utilisateur est correctement rediriger sur la page index
14.04	OK	Aucun	- Le formulaire d'édition des scores s'affiche correctement
14.05	OK	Aucun	- Les scores ce sont correctement enregistrer avec comme valeur 0
14.06	OK	Aucun	- Les scores ce sont correctement enregistrer avec comme valeur 0
14.07	OK	Aucun	- Les score ce sont correctement enregistrer - L'utilisateur est correctement rediriger sur la page index



Cérémonie de fin - KidsGames 2017 | Evaux

## CONCLUSION

Ce projet m'a permis de découvrir le fonctionnement du développement d'une application de sa conception à son implémentation. N'ayant jamais réalisé un projet de cette envergure auparavant, j'ai pu me rendre compte du travail demandé pour faire la conception, la réalisation, mais surtout la documentation.

### MON AVIS SUR LE TPI

Un des défauts du TPI que je pourrais relever, est que la documentation et le journal de bord occupent une trop grande place dans la pondération de la note. Mais ayant connaissance du sujet et des outils que j'utilise, j'ai pu assez rapidement finir le code pour m'attaquer à la documentation.

En général c'est une excellente expérience qui demande rigueur et concentration pendant 10 jours.



J'ai passé pas mal de temps à réaliser l'algorithme qui calcule les points selon les résultats obtenus lors des matchs. C'est la première fois que je me suis détaché de l'ordinateur pour réfléchir au problème sur une feuille. Une fois fini, le transcrire en algorithme s'est fait assez facilement et sans grande difficulté.

## LES POINTS POSITIFS

Je suis content du résultat de mon projet que ce soit au niveau du code ou de l'interface. C'était important pour moi que le site s'adapte sur smartphone et tablette, car pendant le camp, les coachs et arbitres seront principalement sur leur téléphone pour enregistrer les scores.

Un autre point positif est le respect du planning. Du premier au dernier jour j'ai suivi le planning sans décalage ce qui m'a permis de rester organisé et de ne pas traîner tout au long de mon projet.

## LES POINTS À AMÉLIORER

J'aurais pu directement commenté mon code au fur et à mesure que je l'écrivais. L'erreur que j'ai faite a été d'écrire tout le code, de le tester, puis d'écrire les commentaires. J'ai perdu une demi-journée à écrire tous les commentaires.

Une fois les commentaires écrits, coder la suite a été beaucoup plus facile pour moi, car je savais exactement ce que faisaient mes fonctions.

J'ai beaucoup apprécié le fait de savoir que ce projet me servira réellement lors des Kidsgames. Effectivement, après tous les tests effectués durant cette documentation, je pense que mon application fonctionnera entièrement pour l'édition 2018 des KidsGames.



Match de Kinball - KidsGames 2017 | Eaux

# LIENS ET RÉFÉRENCES

---

Dans cette section de la documentation, vous trouverez toutes les références et liens utilisés pour la réalisation du projet.

## LIENS

---

Le template utilisé pour l'application web: <http://demo.graygrids.com/#LightWave>

## PERSONNES RESSOURCES

---

M. Pascal Bonvin m'a aidé à créer les fonctions prepare statement avec les requêtes statiques.

M. Alec Beney m'a appris comment créer une documentation technique et comment la structurer.

Mme. Katia Mota m'a appris à créer un manuel d'utilisateur. Elle m'a également expliqué quel langage utiliser avec les futurs utilisateurs, ainsi que la manière d'afficher les images et expliquer les étapes.

M. Florent Beney m'a aidé à résoudre quelques petits problèmes d'affichages tout au long du TPI.

- Le nom du terrain ne s'affichait pas sur la page day
- La page editScore n'affichait pas les équipes

Merci à M. Foti et M. Fehr de m'avoir suivi tout au long de mon travail.

## DOCUMENTS

---

Pour réaliser le projet, je me suis servis de tout les projets que j'ai réalisé tout au long de l'année durant les ateliers et les modules.