

Gestion d'équipe de Football

Documentation technique



CFPT Informatique
Travail Pratique Individuel
Session mai 2023

Candidat : Gabriel Martin
Maitre d'apprentissage : Mme. Terrier
Experts : Jean-Pierre Garnier & Thomas Tetart

Table des matières

1. Versions	3
2. Introduction	3
3. Résumé du cahier des charges.....	3
3.1 Organisation	3
3.2 But de l'application	4
3.3 Fonctionnalité à réaliser	4
3.4 Matériel et logiciels nécessaires.....	5
3.5 Livrables	5
4. Méthodologie	6
1. S'informer	6
2. Planifier	6
3. Décider	6
4. Réaliser	6
5. Contrôler.....	6
6. Évaluer	7
5. Planification.....	7
5.1 Backlog.....	7
5.2 Planning prévisionnel	16
5.3 Planning réel.....	17
6. Librairies et outils externes.....	18
6.1 Git.....	18
6.2 Visual Studio Code	18
6.3 WSL 2 (Ubuntu).....	18
6.4 FullCalendar	19
6.5 JQuery	20
6.6 Composer	20
6.7 Pecee Simple Router.....	21
6.8 Autoload	21
7. Analyse	22

7.1 Analyse de l'existant	22
7.2 Structure du projet	23
7.3 Base de données.....	24
7.4 Fonctionnalités majeures.....	25
7.5 Exemple de code	28
8. Plan de test et tests	29
8.1 Périmètre des tests.....	29
8.2 Équipement de test.....	29
8.3 Scénarios de test	30
8.4 Évolution des tests	38
8.5 Échec des tests et raisons	39
9. Conclusion	41
9.1 Difficultés rencontrées	41
9.2 Planification	41
9.3 Améliorations possibles.....	42
9.4 Bilan personnel.....	43
10. Remerciements	43
11. Bibliographie	43
12. Table des illustrations	44
13. Annexes	44

1. Versions

N° de version	Date	Auteur	Changements apportés
1.0	21.03.2023	Gabriel Martin	Version finale de la documentation du TPI finale du document pour le rendu du TPI

2. Introduction

Voici la documentation technique du projet *Drafteam* réalisé dans le cadre du *Travail pratique individuel* (TPI) débutant le jeudi 27 avril 2023 et se terminant le mercredi 17 mai 2023. Ce travail a pour but de valider ma formation CFC effectuée au *Centre de Formation Professionnelle Technique* du Petit-Lancy à Genève.

Cette application web a pour but de gérer des équipes de football. Elle permet la création d'équipe, l'ajout de joueurs, la gestion d'entraînements et de matchs. Le tout peut être géré par un entraîneur qui a la possibilité de voir les absences et de mettre à jour lui-même le calendrier de l'équipe.

3. Résumé du cahier des charges

3.1 Organisation

Les informations qui vont suivre sont extraite du cahier des charges.

Élève	Maître d'apprentissage
Gabriel Martin gabriel-henri.mrtn@eduge.ch	Anne Terrier anne.terrier@edu.ge.ch

Experts	
Jean-Pierre Garnier jpg@gsinfo.ch	Thomas Tetart thomas.tetart@laposte.net

3.2 But de l'application

Le but de ce projet est de proposer une application web dédiée aux clubs de football. Cette application de gestion d'équipes vous propose une interface simple permettant de créer et d'organiser vos équipes et vos entraînements.

Afin d'utiliser l'application, l'entraîneur doit tout d'abord se créer son compte en ligne. Cela étant fait, il se connecte et accède à différentes fonctionnalités :

- Création d'une équipe, ajout de joueurs ou de personnes faisant partie du staff grâce à leurs e-mails.
- Création des championnats auxquels participe l'équipe
- Création des événements (entraînement par exemple) et envoi d'invitations

Les joueurs reçoivent, de leur entraîneur, une invitation et un mot de passe par défaut. Une fois connectés à l'application, ils peuvent :

- Configurer leur profil
- Annoncer leurs présences ou leurs absences aux événements auxquels ils ont été invités

Le site web est développé en HTML/CSS/PHP/JS

Les données sont stockées dans une base de données relationnelle MariaDB.

3.3 Fonctionnalité à réaliser

Visiteurs

Un visiteur accède à la page d'accueil du site. Il peut créer un compte (s'il est entraîneur et qu'il souhaite utiliser l'application pour gérer son équipe) ou se connecter.

Entraîneurs et administrateurs

Dans ce chapitre le terme entraîneur inclut également le terme administrateur.

Une fois son compte créé, l'entraîneur dispose des fonctionnalités administrateur. Il peut :

- Créer une équipe.
- Ajouter des sportifs à son équipe (joueurs ou personnels encadrants). Il doit renseigner au minimum, le nom, le prénom, la fonction (via une liste déroulante), l'email et un mot de passe par défaut. Les informations de compte seront communiquées au sportif afin qu'il puisse se connecter dans l'application.
- Créer des championnats.
- Ajouter des événements. Pour chaque événement, le nom, la date, l'heure de début, l'heure de fin ainsi que le type d'événement (liste déroulante) et le lieu (liste déroulante) doivent être renseignés. Les champs image et score sont facultatifs. Ils permettent d'ajouter des informations complémentaires.

- Inviter, pour chaque événement créé, les membres de l'équipe. Il pourra choisir si l'invitation est envoyée à toutes les personnes de l'équipe, aux joueurs seulement ou aux membres encadrants seulement.
- Visualiser les confirmations de présences ou d'absences avec les commentaires liés.
- Ajouter une information sur le résultat de l'événement en utilisant le champ score.
- Visualiser, modifier ou supprimer les équipes, sportifs, championnats et événements liés à l'équipe dont il fait partie.
- Il est le seul à pouvoir modifier le rôle d'un sportif.

Joueurs

Le joueur reçoit les informations de connexion de son entraîneur. Lorsqu'il se connecte sur le site, il peut :

- Compléter ses informations personnelles ou les modifier. Il ne peut pas changer son rôle.
- Voir les informations de son équipe, des championnats auxquels il participe, les autres membres de son équipe.
- Voir les événements futurs auxquels il est invité. Il peut confirmer sa présence ou dire qu'il sera absent. En cas d'absence, il devra justifier de la raison de son absence.
- Voir les événements passés. Dans ce cas, il ne pourra plus indiquer sa présence mais verra le résultat qui aura été mis par l'entraîneur.

3.4 Matériel et logiciels nécessaires

- Un PC standard école, un écran
- Windows 10
- Visual Studio Code
- Suite Office
- Suite AMP à choix
- Outil de versioning (Type Git)
- Outil de création de maquettes (outil proposé Evolus Pencil : <https://pencil.evolus.vn/>)

3.5 Livrables

Le candidat doit livrer à son chef de projet et aux deux experts :

- Une planification initiale dès le premier jour
- Un exemplaire PDF de son mémoire avec annexe
 - Documentation technique
 - Manuel utilisateur
 - Les rapports de tests
 - Code source
- Le résumé du TPI
- Son journal de bord

Le candidat doit livrer à son chef de projet :

- Un accès aux sources du projet

4. Méthodologie



Figure 1 : Image qui représente les méthodologies

Pour le bon fonctionnement de ma gestion de projet, j'ai utilisé la méthode de planification en 6 étapes. Voici le schéma suivi étape par étape tout au long de mon TPI.

1. S'informer

La première étape consiste à prendre conscience des tâches à réaliser et à s'informer sur le problème ou le projet en question. Il s'agit de collecter systématiquement des informations pertinentes, d'identifier les parties prenantes et de comprendre les enjeux du projet.

2. Planifier

La deuxième étape consiste à planifier le projet. Cela implique de déterminer les objectifs à atteindre, d'établir des tâches, d'estimer le temps nécessaire pour les effectuer de déterminer les ressources nécessaires et d'établir un planning.

3. Décider

Une fois les informations récupérées, les tâches planifiées, il faut décider de la manière dont on va réaliser ces dernières. Pour cela, il faut classer les tâches par ordre d'importance afin de réaliser les tâches les plus importantes en premier.

4. Réaliser

La quatrième étape consiste à la réalisation des tâches. Il faut donc suivre le planning créé dans l'étape « Planifier » ainsi que les décisions pour suivre les tâches dans le bon ordre.

5. Contrôler

Une fois la tâche terminée, il faut surveiller le progrès, que ce qui a été effectué fonctionne correctement selon ce qui a été demandé. Pour cela, il faut effectuer un protocole de test pour voir si la fonctionnalité ajoutée est bien fonctionnelle.

6. Évaluer

La dernière étape consiste à évaluer le projet. Cela implique de comparer les résultats obtenus aux objectifs établis, de mesurer l'efficacité des solutions mises en place et de déterminer les mesures à prendre pour améliorer les résultats si nécessaire.

5. Planification

5.1 Backlog

Pour le suivi de mon backlog, j'ai utilisé Trello afin de mettre à jour l'avancement et la difficulté de mes tâches.

Nom	S1 : Utilisation d'un dépôt Git
Description (User story)	En tant que développeur, je dois créer un dépôt git afin de pouvoir enregistrer, versionner et accéder à toutes les versions de mon site.
Critère d'acceptation	Dépôt Git créé en privé avec un droit d'accès pour moi et pour la maîtresse d'apprentissage.
Priorité	Bloquant

Nom	S2 : Création d'une base de données
Description (User story)	En tant que développeur, je dois créer ma base de données pour mon site en respectant l'UML fourni dans le cahier des charges. Je le crée donc manuellement dans DBeaver.
Critère d'acceptation	La base de données respecte l'UML de l'énoncé et permet à mon projet de fonctionner.
Priorité	Bloquant

Nom	S3 : Implémentation de la Template
Description (User story)	En tant que développeur, je dois incorporer la Template dédié pour mon site. Je dois m'assurer que le design soit présentable et s'incorpore bien pour me focaliser sur le back-end
Critère d'acceptation	Le design respecte les consignes du cahier des charges et possède les éléments requis comme une navigation, le contenu spécifique à la page, etc.
Priorité	Bloquant

Nom	S4 : Connexion à la base de données
Description (User story)	En tant que développeur, je dois avoir une connexion entre mon site et la base de données
Critère d'acceptation	Des données peuvent être insérées, récupérées, supprimées ou modifiées dans la base de données grâce à mes fonctions PHP
Priorité	Bloquant

Nom	S5 : Implémentation du routage et de la structure du code
Description (User story)	En tant que développeur, je dois avoir un code propre, bien ordonné et facile d'utilisation.
Critère d'acceptation	La navigation entre les pages se fait correctement grâce Simple Router, les class sont bien reliées grâce à autoload et la structure du code est en MVC.
Priorité	Bloquant

Nom	S6 : Modification du design
Description (User story)	En tant que développeur, je dois avoir un design qui s'adapte aux fonctionnalités et à l'ajout de page.
Critère d'acceptation	Chaque page ajoutée à un design en fonction de ce qu'elle fait.
Priorité	Critique

Nom	S7 : Inscription pour l'entraîneur
Description (User story)	En tant qu'entraîneur non identifié, je dois pouvoir me créer un compte entraîneur grâce à mon email, nom, prénom, mot de passe et mon poste dans l'équipe
Critère d'acceptation	Le test 1.1 passe.
Priorité	Bloquant

Nom	S8 : Gestion de championnats
Description (User story)	En tant qu'entraîneur identifié, je dois pouvoir créer, modifier et supprimer un championnat.
Critère d'acceptation	Les tests 2.1 et 2.2 passent.
Priorité	Critique

Nom	S9 : Gestion d'équipe
Description (User story)	En tant qu'entraîneur identifié, je dois pouvoir créer, modifier et supprimer l'équipe dont je suis l'entraîneur.
Critère d'acceptation	Les tests 3.1 et 3.2 passent.
Priorité	Bloquant

Nom	S10 : Gestion de sportif
Description (User story)	En tant qu'entraîneur identifié, je dois pouvoir créer des comptes pour les joueurs et staff de mon équipe. Je dois pouvoir aussi les modifier et supprimer.
Critère d'acceptation	Le test 4.1 passe.
Priorité	Bloquant

Nom	S11 : Connexion
Description (User story)	En tant qu'utilisateur non identifié et possédant un compte, je peux me connecter à mon compte en entrant mes informations de connexion.
Critère d'acceptation	Le test 5.1 passe.
Priorité	Bloquant

Nom	S12 : Déconnexion
Description (User story)	En tant que sportif identifié, je peux me déconnecter de mon compte.
Critère d'acceptation	Le test 6.1 passe.
Priorité	Critique

Nom	S13 : Modification du profil
Description (User story)	En tant que sportif identifié, je peux modifier mes informations sur ma page profil comme mon nom, prénom, mot de passe, photo, numéro de maillot etc.
Critère d'acceptation	Le test 7.1 passe
Priorité	Critique

Nom	S14 : Gestions des événements
Description (User story)	En tant qu'entraîneur, je dois pouvoir créer des événements, les modifier et les supprimer.
Critère d'acceptation	Les tests 8.1 et 8.2 passe.
Priorité	Bloquant

Nom	S15 : Affichage des événements
Description (User story)	En tant que sportif identifié, je dois pouvoir voir les événements de mon équipe sur le site.
Critère d'acceptation	Le test 9.1 passe.
Priorité	Bloquant

Nom	S16 : Invitations aux événements
Description (User story)	En tant qu'entraîneur, je dois pouvoir inviter mes joueurs et mon staff aux événements que j'ai créés.
Critère d'acceptation	Le test 10.1 passe.
Priorité	Bloquant

Nom	S17 : Confirmations de présence
Description (User story)	En tant que joueur et staff, je peux confirmer ma présence aux événements dans lesquels l'entraîneur de mon équipe m'a invité.
Critère d'acceptation	Le test 11.1 passe.
Priorité	Critique

Nom	S18 : Confirmation d'absence avec un commentaire
Description (User story)	En tant que joueur et staff, je peux répondre absent à un événement en y ajoutant un commentaire.
Critère d'acceptation	Le test 12.1 passe.
Priorité	Critique

Nom	S19 : Visualisation des présences et commentaire
Description (User story)	En tant qu'entraîneur, je peux visualiser les présences de mon équipe aux événements que j'ai créés et les commentaires des absents.
Critère d'acceptation	Le test 13.1
Priorité	Critique

Nom	S20 : Gestion des scores pour les matchs terminés
Description (User story)	En tant qu'entraîneur, je peux ajouter un score aux matchs qui sont terminés.
Critère d'acceptation	Le test 14.1
Priorité	Important

Nom	S21 : Modification du rôle d'un sportif
Description (User story)	En tant qu'entraîneur, je peux modifier les rôles des membres de mon équipe.
Critère d'acceptation	Le test 15.1
Priorité	Critique

Nom	S22 : Affichage des informations de l'équipe
Description (User story)	En tant que sportif identifié, je peux voir les informations relatives à mon équipe
Critère d'acceptation	Le test 16.1
Priorité	Critique

Nom	S23 : Affichage des membres de l'équipe
Description (User story)	En tant que sportif identifié, je peux voir les membres de mon équipe.
Critère d'acceptation	
Priorité	Critique

Nom	S24 : Affichage des informations championnat
Description (User story)	En tant que sportif identifié, je peux voir les informations relatives à mon championnat.
Critère d'acceptation	
Priorité	Critique

Voici donc à quoi ressemble mon Trello pendant la réalisation de mon TPI. J'ai pris comme exemple le deuxième jour, mardi 2 mai :

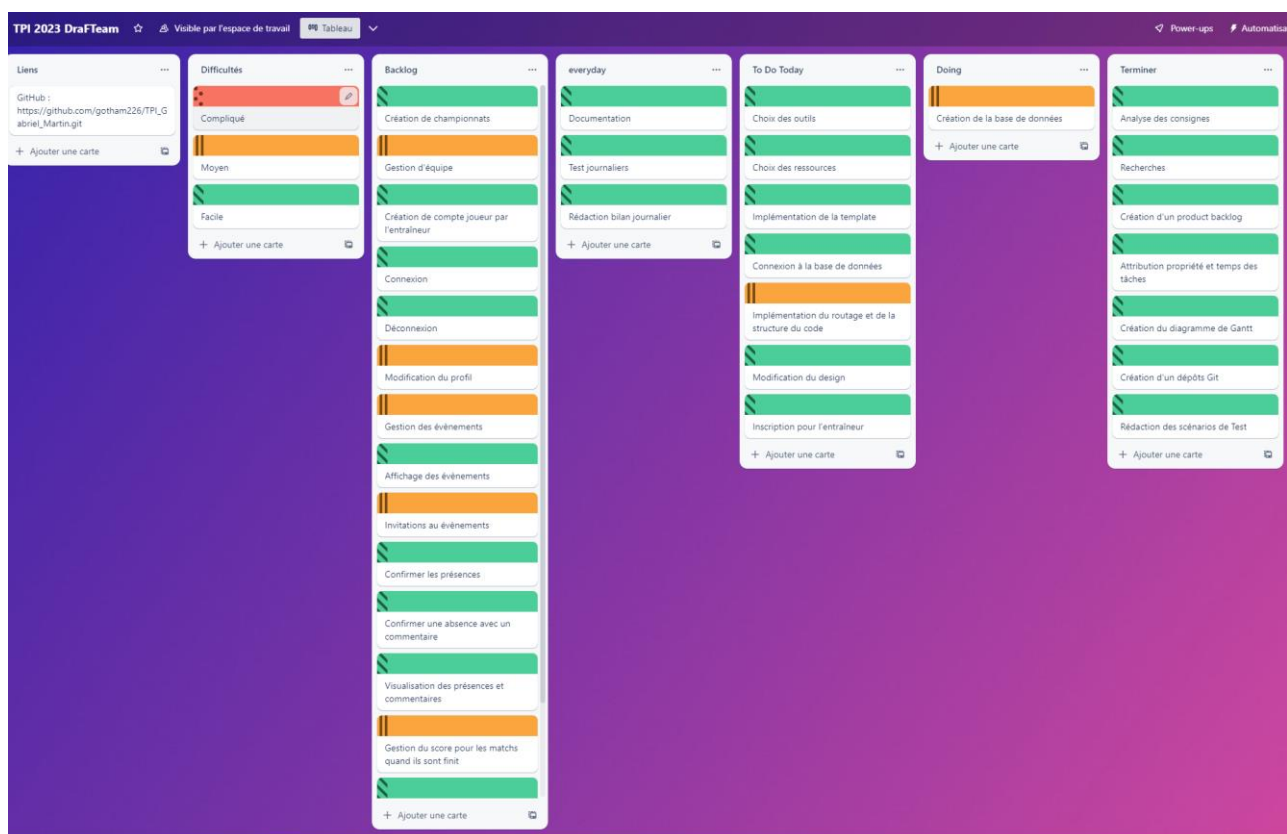


Figure 2 : Exemple du Trello du 2 mai

5.2 Planning prévisionnel

DraFTeam

Site de gestion d'équipe de football
Gabriel-Henri Martin

Début du projet : jeudi, 27. avril 2023

Maître d'apprentissage: Mme Terrier

		27 avr 2023	2 mai 2023	3 mai 2023	4 mai 2023	6 mai 2023	9 mai 2023	10 mai 2023	11 mai 2023	15 mai 2023	16 mai 2023	17 mai 2023	
TÂCHES	Nombre d'heures planifié	8.00	8.00	8.00	8.00	8.00	8.00	8.00	8.00	8.00	8.00	8.00	
Introduction au projet	1.00												
Analyse des consignes	0.30	0.30											0.30
Recherches	0.30	0.30											0.30
Planification	4.30												
Création d'un product backlog	1.00	1.00											1.00
Attribution de la propriété et du temps des tâches	1.00	1.00											1.00
Création du diagramme de Gantt	1.00	1.00											1.00
Rédaction des scénarios de test	1.30	1.30											1.30
Décisions	0.20												0.00
Choix des outils	0.10		0.10										0.10
Choix des ressources	0.10		0.10										0.10
Réalisation	73.55												
S1 Création d'un dépôt Git	0.15	0.15											0.15
S2 Création de la base de données	2.00	1.00	1.00										2.00
S3 Implémentation de la template	0.30		0.30										0.30
S4 Connexion à la base de données	0.10		0.10										0.10
S5 Implémentation du routage et de la structure du code	1.30		1.30										1.30
S6 Modification du design	4.00		0.30	1.30	0.30	0.45	0.30	0.15					4.00
S7 Inscription pour l'entraîneur	1.30		1.30										1.30
S8 Création de championnats	1.00			1.00									1.00
S9 Gestion d'équipe	4.00			1.15	2.45								4.00
S10 Création de compte joueur par l'entraîneur	2.00				2.00								2.00
S11 Connexion	1.30			1.30									1.30
S12 Déconnexion	0.30					0.30							0.30
S13 Modification du profil	2.00					2.00							2.00
S14 Gestion des événements	4.00						4.00						4.00
S15 Affichage des événements	2.00					2.00							2.00
S16 Invitation aux événements	2.00							2.00					2.00
S17 Confirmation de présence	2.00							2.00					2.00
S18 Confirmation d'absence avec un commentaire	2.00								2.00				2.00
S19 Visualisation des présences et commentaires	1.00								1.00				1.00
S20 Gestion des score pour les matchs terminés	2.00								0.15	1.45			2.00
S21 Modification du rôle d'un sportif	0.30										0.30		0.30
S22 Affichage des informations équipe	2.00										2.00		2.00
S23 Affichage des membres de l'équipe	1.30											1.30	1.30
S24 Affichage des informations championnat	2.00											2.00	2.00
Documentation	32.00	1.00	1.45	2.00	2.00	2.00	2.45	3.00	4.00	2.50	3.35	7.05	32.00
Contrôles	5.30												
Test journaliers	5.30		0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.40	0.40	0.40	5.30
Evaluation	2.45												
Rédaction du bilan journalier	2.45	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	2.45
Temps total	88.00												

Figure 3 : Planning prévisionnel

5.3 Planning réel

DraFTeam

Site de gestion d'équipe de football
Gabriel-Henri Martin

Début du projet : jeudi, 27. avril 2023

Maître d'apprentissage: Mme Terrier

TÂCHES	Nombre d'heure planifié	27 avr 2023	2 mai 2023	3 mai 2023	4 mai 2023	8 mai 2023	9 mai 2023	10 mai 2023	11 mai 2023	15 mai 2023	16 mai 2023	17 mai 2023
Introduction au projet	1:00											
Analyse des consignes	0:30	0:30										
Recherches	0:30	0:30										
Planification	4:30											
Création d'un product backlog	1:00	1:00										
Attribution de la propriété et du temps des tâches	1:00	1:00										
Création du diagramme de Gantt	1:00	1:00										
Rédaction des scénarios de test	1:30	1:30										
Décisions	0:20											
Choix des outils	0:10		0:10									
Choix des ressources	0:10		0:10									
Réalisation	73:55											
S1 Création d'un dépôt Git	0:15	0:15										
S2 Création de la base de données	2:00	1:00	1:00									
S3 Implémentation de la template	0:30		0:30									
S4 Connexion à la base de données	0:10		0:10									
S5 Implémentation du routage et de la structure du code	1:30		1:30									
S6 Modification du design	5:00		0:30	1:30	0:30	0:45	0:30	1:15				
S7 Inscription pour l'entraîneur	1:30		1:30									
S8 Gestion de championnats	2:30		1:00	1:00	0:30							
S9 Gestion d'équipe	2:30			1:45	0:45							
S10 Gestion de sportif	3:00				2:00	1:00						
S11 Connexion	1:30			1:30								
S12 Déconnexion	0:30			0:30								
S13 Modification du profil	2:30					2:30						
S14 Gestion des événements	4:30					1:00	3:30					
S15 Affichage des événements	2:00						2:00					
S16 Invitation aux événements	1:30							1:30				
S17 Confirmation de présence	1:30							1:30				
S18 Confirmation d'absence avec un commentaire	2:00							2:00				
S19 Visualisation des présences et commentaires	2:00								2:00			
S20 Gestion des score pour les matchs terminés	2:30								2:30			
S21 Modification du rôle d'un sportif	0:30					0:30						
S22 Affichage des informations équipe	1:00				1:00							
S23 Affichage des membres de l'équipe	1:30				1:30							
S24 Affichage des informations championnat	0:30					0:30						
Documentation	31:00	1:00	0:45	1:00	1:00	1:00	1:15	1:00	2:45	7:05	7:05	7:05
Contrôles	5:30											
Test journaliers	5:30		0:30	0:30	0:30	0:30	0:30	0:30	0:30	0:40	0:40	0:40
Evaluation	2:45											
Rédaction du bilan journalier	2:45	0:15	0:15	0:15	0:15	0:15	0:15	0:15	0:15	0:15	0:15	0:15
Temps total	88:00											

Figure 4 : Planning réel

6. Bibliothèques et outils externes

6.1 Git

Git est un logiciel de gestion de versions distribué, développé par Linus Torvalds, le créateur du noyau Linux. Pour assurer le suivi des différentes versions de mon application, j'ai opté pour l'utilisation d'un dépôt distant sur la plateforme GitHub. Cela me permet de stocker mon code source en ligne et de suivre les changements apportés à chaque version de mon application.

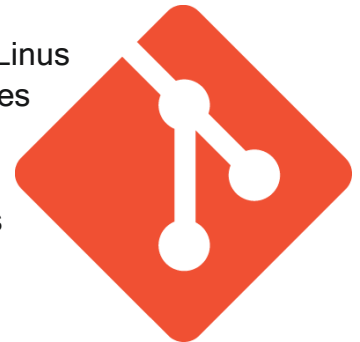


Figure 5 : Logo Git

6.2 Visual Studio Code

Visual Studio Code est un environnement de développement intégré (IDE) polyvalent, simple à utiliser et doté d'une large gamme de fonctionnalités. Il permet de programmer dans divers langages de programmation et sa modularité est renforcée par son système d'extension. Personnellement, j'ai utilisé Visual Studio Code pour développer mon application et cet IDE m'a été extrêmement utile pendant la phase de développement.

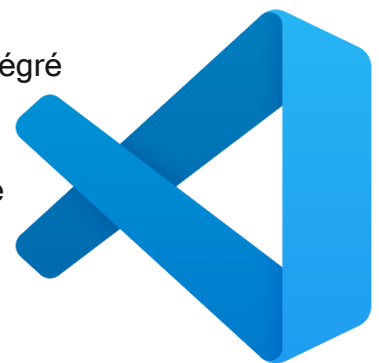


Figure 6 : Logo Visual Studio Code

6.3 WSL 2 (Ubuntu)

WSL 2 est une fonctionnalité de Windows qui permet de créer un environnement Linux à l'intérieur de Windows, ce qui équivaut à avoir une machine virtuelle Linux intégrée dans la machine Windows. Cela permet d'exécuter des applications Linux sur Windows sans avoir besoin d'une machine virtuelle distincte. Mon application ainsi que ma base de données ont donc tourné sur un serveur en local.

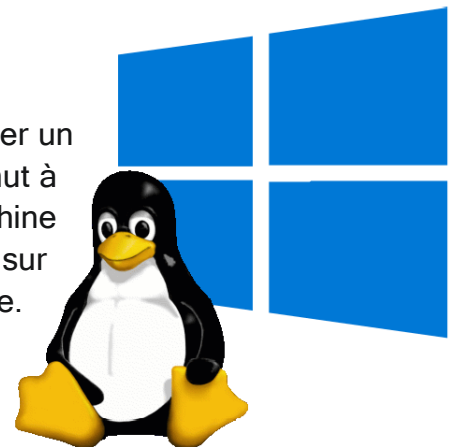


Figure 7 : Logo Linux avec logo Windows

6.4 FullCalendar

FullCalendar est une bibliothèque JavaScript open-source permettant de créer des calendriers interactifs sur une page web. J'ai utilisé FullCalendar dans mon application de gestion d'équipe de football pour afficher les matchs et les événements importants dans un calendrier facile à naviguer. Grâce à FullCalendar, j'ai pu afficher les matchs, les entraînements et divers événements, puis permettre aux utilisateurs de cliquer dessus pour obtenir plus d'informations. L'utilisation de FullCalendar a grandement amélioré l'expérience utilisateur de mon application en offrant une vue d'ensemble de tous les événements à venir et en permettant aux utilisateurs de planifier en conséquence.



Voici le code qui ma servit à créer un calendrier :

Figure 8 : Logo FullCalendar

```
// Ajout d'un écouteur d'événement qui se déclenche lorsque le contenu du document est chargé
document.addEventListener('DOMContentLoaded', function() {

    // Récupération de l'élément HTML avec l'id "calendar"
    var calendarEl = document.getElementById('calendar');

    // Création d'une instance de FullCalendar en passant l'élément HTML récupéré précédemment comme paramètre
    // ainsi que d'autres options telles que la vue initiale, l'emplacement du fichier PHP contenant les événements,
    // et d'autres configurations pour les différentes vues du calendrier
    calendar = new FullCalendar.Calendar(calendarEl, {

        // Pour afficher la vue initial (par mois)
        initialView: 'dayGridMonth',
        events: '/agendaEvent', // récupère les events

        // Affichage des boutons en haut du calendrier pour naviger de mois en mois ou semaine en semaine
        headerToolbar: { center: 'dayGridMonth,timeGridWeek' },
        views: { // Les différentes vues :
            dayGrid: {
                contentHeight: 1000
            },
            timeGrid: {
                // Les différentes vues :
                week: {
                    contentHeight: 1000
                },
                day: {
                    // Les différentes vues :
                }
            }
        }
    });
```

Figure 9 : Code pour la création d'un calendrier

Appel de la bibliothèque FullCalendar :

```
<script src='https://cdn.jsdelivr.net/npm/fullcalendar@6.1.5/index.global.min.js'>
</script>
```

6.5 JQuery

JQuery est une bibliothèque JavaScript utile pour faciliter la manipulation du DOM et la gestion des événements. Dans mon code, l'utilisation de jQuery a simplifié la sélection des éléments HTML et la gestion des événements, ce qui a rendu le code plus court et plus facile à comprendre.

Appel de la bibliothèque JQuery :

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js">
</script>
```



Figure 10 : Logo JQuery

6.6 Composer

Composer est un gestionnaire de dépendances qui m'a permis de gérer efficacement les bibliothèques, les frameworks et les packages dont mon projet avait besoin. Grâce à Composer, j'ai pu automatiser l'installation et la mise à jour de ces dépendances en créant un fichier de configuration nommé "composer.json". En utilisant Packagist, un référentiel de package, Composer a pu trouver et télécharger les packages requis.

L'utilisation de Composer a apporté de nombreux avantages à mon projet. En gérant les dépendances de manière centralisée, j'ai pu éviter les erreurs d'incompatibilité et m'assurer que toutes les bibliothèques et les packages étaient correctement installés. En outre, en permettant une mise à jour automatique des dépendances, Composer facilite la maintenance de mon code. Enfin, en automatisant la gestion des packages, Composer m'a fait gagner du temps et m'a permis de me concentrer sur le développement de mon application.



Figure 11 : Logo Composer

6.7 Pecee Simple Router

Pecee Simple Router est une bibliothèque PHP qui permet la gestion des routes dans une application web. Elle permet de définir des URL pour chaque page ou fonctionnalité de mon site et de rediriger les requêtes HTTP vers la bonne page en fonction de l'URL demandée.

Pecee Simple Router a rendu mon site plus intéressant de plusieurs façons. Tout d'abord, elle permet de créer des URL plus propres, ce qui améliore l'expérience utilisateur. Ensuite, elle facilite la maintenance et l'organisation de mon code en séparant clairement les actions à réaliser en fonction des routes demandées. Enfin, elle améliore les performances de mon site en réduisant la charge de traitement nécessaire pour déterminer la page ou la fonctionnalité à exécuter.

En somme, l'utilisation de Pecee Simple Router m'a aidé à améliorer la qualité globale de mon site web en améliorant l'expérience utilisateur, la maintenance et l'organisation de mon code, et les performances.

Voici un exemple de route pour la page [accueil.php](#) :

```
Router::form('/', [AccueilController::class, 'accueil']);
```

Figure 12 : Code montrant la création d'une route pour la page accueil

6.8 Autoload

L'autoload est une fonctionnalité de PHP qui permet de charger automatiquement les classes nécessaires à l'exécution de mon code, sans que j'aie à les inclure manuellement dans chaque fichier. Cela permet de simplifier le processus de développement, en évitant les erreurs d'inclusion de fichiers et en facilitant la maintenance du code en utilisant des namespace et use.

```
3 namespace drafteam\Models;  
4 use drafteam\Models\database;
```

Figure 13 : Code montrant l'utilisation d'un namespace et d'une classe

Utiliser l'autoload m'a permis de travailler plus efficacement en évitant les tâches répétitives liées à l'inclusion manuelle de classes. Ensuite, elle a facilité la maintenance de mon code en permettant une organisation claire de mes classes dans des dossiers spécifiques. Enfin, elle a permis de réduire le temps de chargement de mes pages en évitant le chargement de classes inutiles.

7. Analyse

7.1 Analyse de l'existant

Mon TPI implique la création d'un site web permettant la gestion d'équipe de football par un entraîneur pour ses joueurs.

Dans ce contexte, je souhaite explorer les avantages et les inconvénients de l'utilisation d'un site web plutôt que d'un outil dédié à la gestion d'un agenda comme l'agenda de Google.

Comparaisons avec l'agenda partagé de Google

Avec la liberté offerte par la programmation web, je trouve que l'agenda Google offre moins de possibilités et d'améliorations possibles que pour un site web. Dans mon cas, le PHP, l'HTML/CSS et le JS permettent de répondre à n'importe quelle demande, que ce soit avec des librairies externes ou du code effectué par moi-même.

En effet, avec l'agenda Google, nous pourrions utiliser quelque chose de déjà fait et nous n'aurions pas besoin de recréer la roue. La manipulation est simple et est intégrée à l'environnement Google, ce qui en fait un outil pratique et simple.

Cependant, avec l'agenda Google, nous sommes vite limités en termes de fonctionnalités. Dans le cas de mon site web, la bibliothèque JavaScript [FullCalendar](#) permet de créer un agenda de manière dynamique et de ne pas repartir de zéro. Ensuite, le PHP m'a permis d'enregistrer mes utilisateurs et événements dans une base de données et l'HTML/CSS m'a permis d'avoir un design unique et propre à mon site.

Pour finir, l'utilisation de mon site web plutôt que de l'agenda de Google favorise grandement la possibilité de fonctionnalités, la gestion d'équipe en tant que telle, les statistiques et l'interaction entre l'entraîneur et les joueurs pour les présences. De plus, les améliorations futures telles qu'un classement pour les championnats seraient une plus-value non négligeable.

7.2 Structure du projet

La structure de mon application a été réalisé en MVC, c'est-à-dire sous la forme : Modèle, Vue Controller.

/index.php

Ce fichier sert à enregistrer un namespace par défaut pour les controllers et à lancer le router.

/.htaccess

Ce fichier redirige toutes les requêtes vers le fichier `index.php`.

/routes.php

Contient toutes les routes de mon application.

/Views/

Contient toutes les pages d'affichages de l'application.

/Models/

Contient toutes les classes models pour les requêtes à la base de données.

/Controllers/

Contient le code qui correspond à chaque route et chacun de ces controller appelle sa vue.

/assets/

Ce dossier contient toutes les images (images du site et image télécharger par l'utilisateur), les fichiers javascript et CSS.

Cette structure m'a été imposé dans le cahier des charges pour ce qui concerne le MVC, j'ai moi-même décidé d'utiliser des routes pour faciliter l'organisation de mon projet et la fluidité de mon site pour l'utilisateur. Même si cette structure ne m'aurait pas été imposée je l'aurais choisi car je trouve cette façon de coder beaucoup plus soigner et organiser.

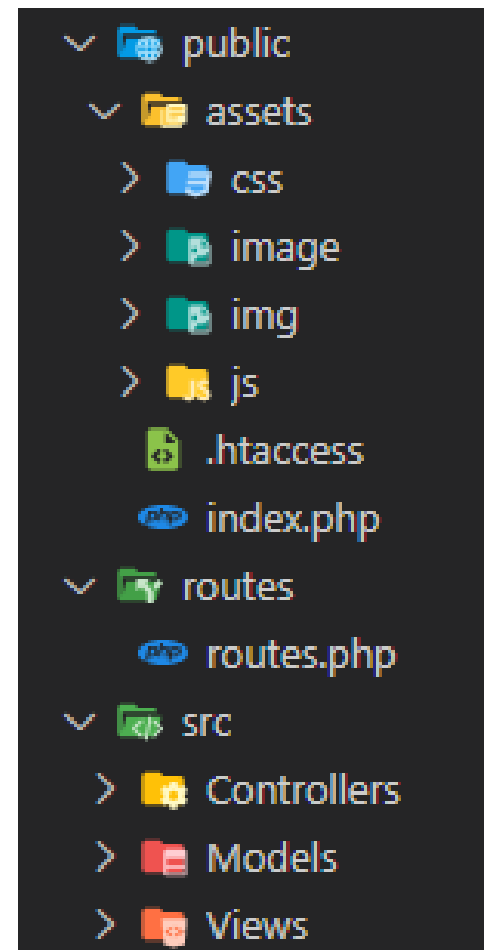


Figure 14 : Image de l'arborescence

7.3 Base de données

Pour la réalisation de Drafteam j'ai utilisé une base de données MySQL afin de stocker les données concernant les utilisateurs, équipes, événement etc. Cette base de données ma été fourni dans mon cahier des charges, néanmoins j'ai dû la modifier tout au long de mon TPI pour qu'elle corresponde à l'attente de l'application.

Base de données du cahier des charges :

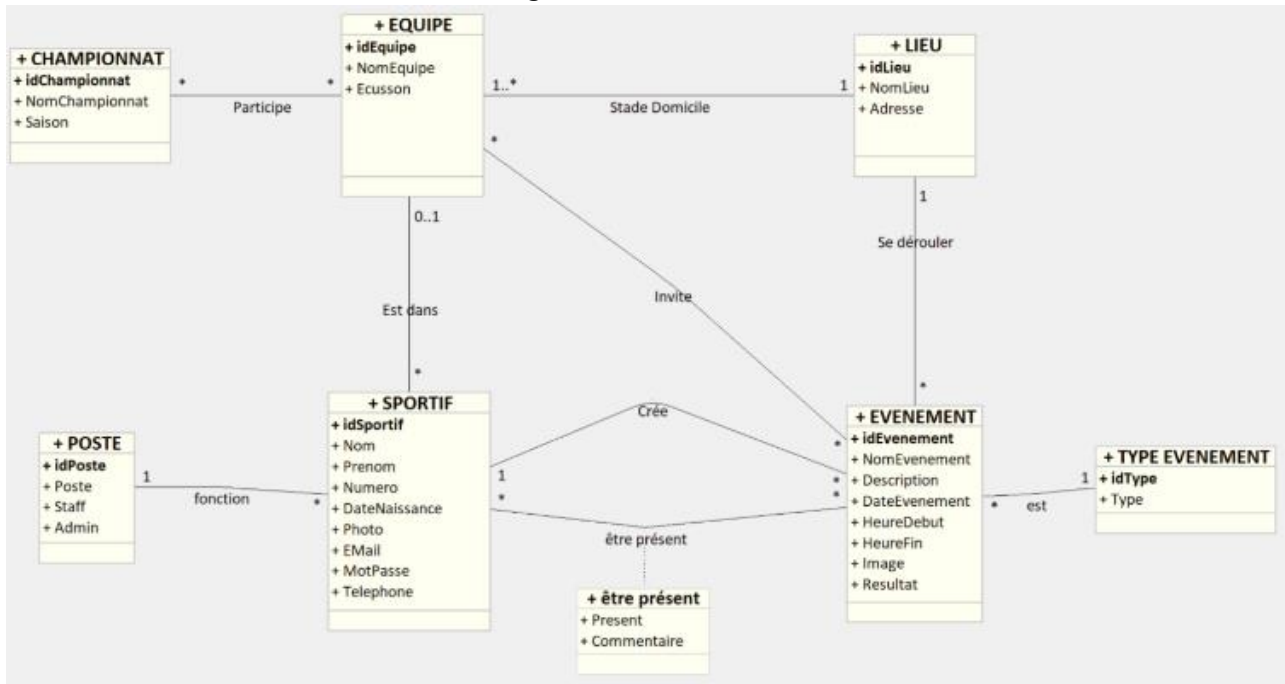


Figure 15 : Base de données du cahier des charges

Base de données final :

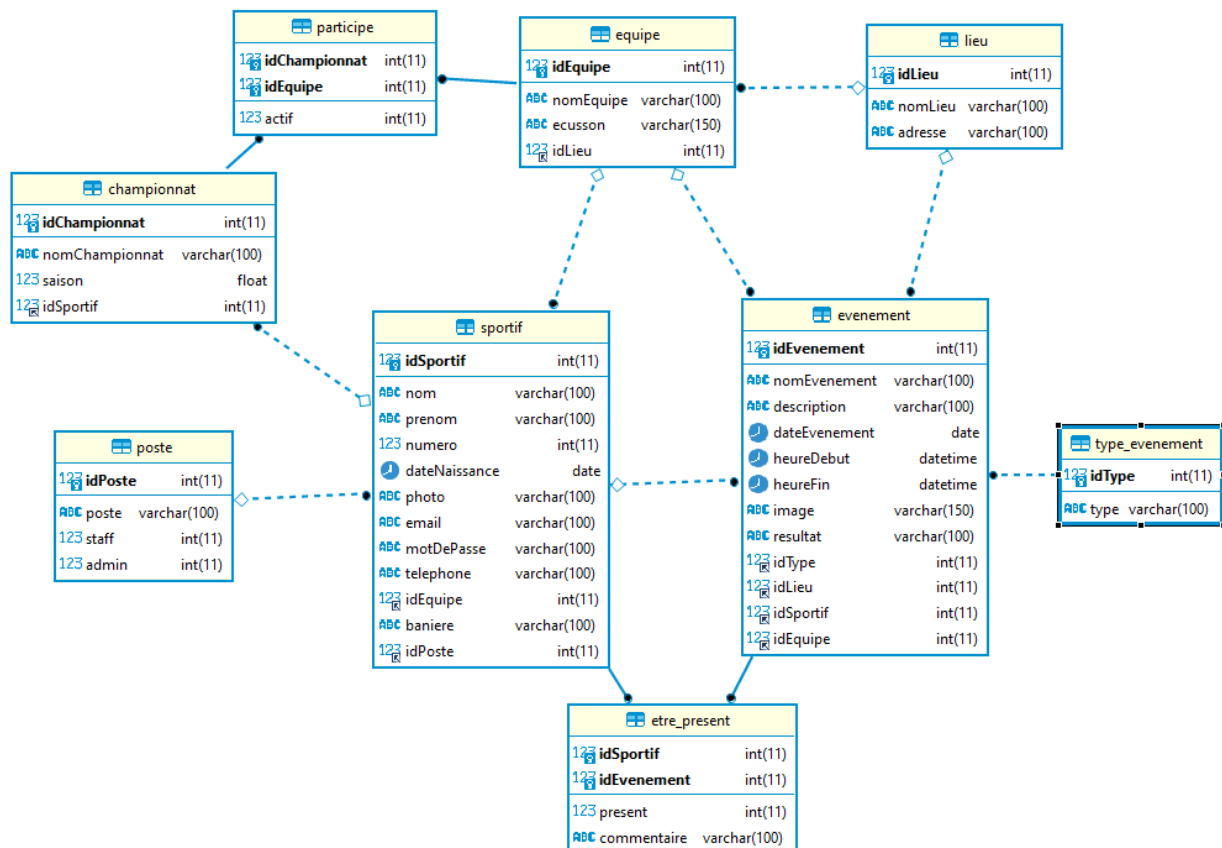


Figure 16 : Base de données final

7.4 Fonctionnalités majeures

1. Système de connexion

Cette fonctionnalité permet à l'entraîneur de s'inscrire et de créer des comptes pour ses joueurs et son staff. Elle permet également à tous les utilisateurs possédant un compte de se connecter.

Inscription

Lorsqu'un entraîneur souhaite se créer un compte pour gérer son équipe il doit pouvoir s'inscrire. L'inscription se passe de la façon suivante : l'entraîneur va rentrer ses données telles que son nom, prénom, email, mot de passe etc. Ensuite si tous les champs sont renseignés, que les deux mots de passe sont identiques, une requête à la base de données est effectuée pour savoir si l'email n'est pas déjà renseigné dans la base de données. Pour finir, on fait une requête d'insertion dans la base de données et l'entraîneur est donc inscrit.

```
$sql = "INSERT INTO sportif(nom, prenom, dateNaissance, photo, email, motDePasse, telephone, idPoste, baniere, idEquipe)
VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?);";
```

Figure 17 : Requête SQL pour l'inscription d'un sportif

Connexion

Lorsqu'un sportif ou entraîneur veut se connecter, il doit renseigner son email et son mot de passe, si les deux sont renseignés et que le mot de passe et l'email correspondent à ceux dans la base de données, il est redirigé sur la page d'accueil en étant connecté.

2. Gestion d'équipe

Cette fonctionnalité permet à l'entraîneur de créer, modifier et supprimer une équipe. Il peut inviter des sportifs (joueurs et staff) à rejoindre son équipe et inscrire l'équipe à des championnats.

Création et affichage d'une équipe

Lorsqu'un entraîneur créer une équipe, il doit renseigner son nom, le stade dans lequel l'équipe joue à domicile et puis enfin son écusson. L'équipe est visible des autres sportif utilisant l'application en se rendant dans le championnat où cette certaine équipe joue.

Voici la requête qui sert à afficher ces équipes participant à un championnat :

```
"SELECT *
FROM equipe e
INNER JOIN participe p ON e.idEquipe = p.idEquipe
INNER JOIN championnat c ON p.idChampionnat = c.idChampionnat
WHERE c.idChampionnat = ? AND p.actif = 1;";
```

Figure 18 : Requête SQL pour l'affichage des équipes participant à un championnat

3. Gestion de championnat

Cette fonctionnalité permet à l'entraîneur de créer, modifier et supprimer des championnats. Il peut également inscrire et désinscrire son équipe aux différents championnats.

Affichage des championnats

N'importe quel sportif peut voir tous les championnats disponibles dans l'application avec une séparation entre les championnats actifs et les autres. L'entraîneur peut voir les championnats qu'il a créé séparément des autres championnats.

Voici la requête qui nous renvoie le championnat actif s'il y en a un :

```
$sql = "SELECT idChampionnat FROM participe WHERE actif = ? AND idEquipe = ?";  
$data = [  
    1,  
    $idEquipe  
];  
$result = database::dbRun($sql, $data)->fetchColumn();  
return $result !== false ? $result : null;
```

Figure 19 : Requête qui renvoie le championnat actif s'il y en a un

4. Gestion des utilisateurs

Cette fonctionnalité permet à l'entraîneur de gérer les utilisateurs. Il peut créer un compte pour ses joueurs et son staff pour qu'ils puissent rejoindre l'équipe et changer leurs rôles si besoin.

Création d'un compte pour un sportif

Un entraîneur doit créer le compte de ses sportifs pour qu'il puisse se connecter par la suite et avoir accès aux informations de leur équipe. L'entraîneur peut donc créer les comptes de ses joueurs/staffs en entrant seulement leur email et leur poste car un mot de passe aléatoire est généré par l'application. Il a plus qu'à copier le mot de passe et l'envoyer à son joueur/staff pour qu'il puisse se connecter.

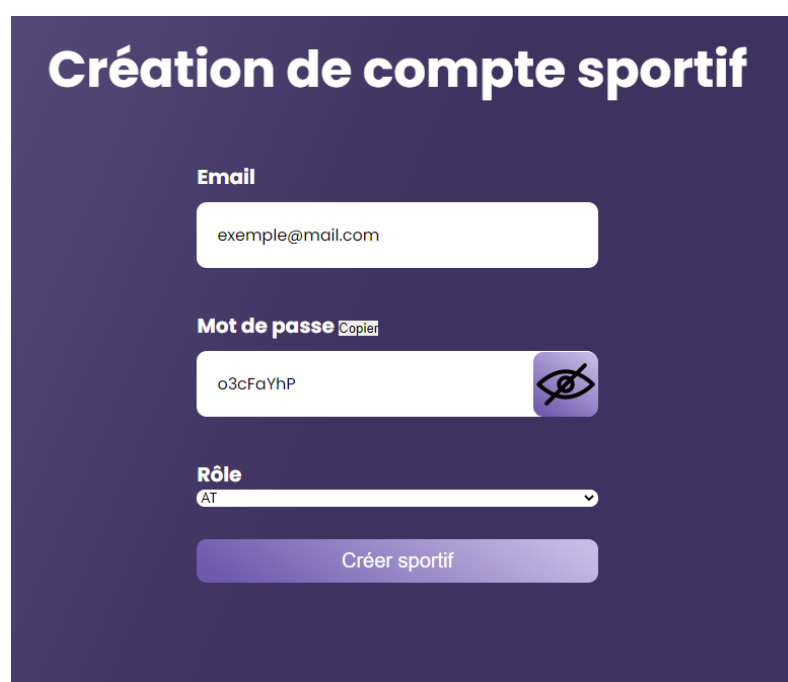


Figure 20 : Capture d'écran du formulaire d'ajout de joueurs

5. Système d'événements

Cette fonctionnalité permet à l'entraîneur de créer, modifier et supprimer des événements. Il peut inviter les sportifs de son équipe aux événements et ces derniers peuvent répondre présents ou absents avec un commentaire d'excuse.

Affichage des présences

Lorsque l'entraîneur se rend sur la page d'un événement, il peut consulter les présences des joueurs/staffs qu'il a invités.

- ✓ Quand un sportif a répondu présent il y a un petit vu vert à côté de son prénom.
- ✗ S'il a répondu absent, il y a une petite croix rouge qui est affichée.
- 🗨️ Quand le joueur a répondu absent, il a dû mettre une raison à son absence. Cette icône s'affiche à côté des joueurs absents. L'entraîneur peut voir les commentaires en cliquant dessus.
- 🕒 Si le sportif n'a toujours pas répondu, il y a une petite horloge orange sur sa carte.



Figure 21 : Capture d'écran de la page d'un événement

7.5 Exemple de code

Ce code JavaScript gère l'interaction avec un élément de type "dropzone" et un bouton d'upload sur une page web. Voici les différentes fonctionnalités implémentées :

1. Lorsqu'on clique sur la zone de dépôt (dropzone), la fenêtre de sélection de fichiers s'ouvre grâce à l'événement "click" sur l'élément "input".
2. Lorsqu'on survole la zone de dépôt avec un fichier en cours de drag-and-drop, le comportement par défaut du navigateur est empêché grâce à l'événement "dragover", et la couleur de fond de la zone de dépôt est modifiée pour donner un effet visuel à l'utilisateur.
3. Lorsque le curseur quitte la zone de dépôt, l'événement "dragleave" réinitialise le style de la zone de dépôt et empêche le comportement par défaut du navigateur.
4. Lorsqu'un fichier est sélectionné via le bouton de sélection de fichiers, l'événement "change" est déclenché sur l'élément "input". Si le fichier est une image, son nom est affiché dans l'élément "filename" et le bouton d'upload est activé.

```
let dropzone = document.getElementById("dropzone");
let uploadButton = document.getElementById("upload-button");
let filename = document.getElementById("filename");
let input = document.getElementById("image");

// Ouvrir la fenêtre de sélection de fichiers en cliquant sur
dropzone.addEventListener("click", function() {
    input.click();
});

// Empêcher le comportement par défaut du navigateur lorsqu'on
dropzone.addEventListener("dragover", function(event) {
    event.preventDefault();
    dropzone.style.backgroundColor = "lightgray";
});

dropzone.addEventListener("dragleave", function(event) {
    event.preventDefault();
    dropzone.style.backgroundColor = "";
});

// Gérer la sélection de fichiers
input.addEventListener("change", function() {
    let file = input.files[0];
    if(file && file.type.match(/image.*/)) {
        filename.innerText = file.name;
        uploadButton.disabled = false;
    }
});

// Gérer le dépôt de fichiers
dropzone.addEventListener("drop", function(event) {
    event.preventDefault();
    dropzone.style.backgroundColor = "";
    let files = event.dataTransfer.files;
    for(let i = 0; i < files.length; i++) {
        let file = files[i];
        if(file.type.match(/image.*/)) {
            filename.innerText = file.name;
            input.files = files;
            uploadButton.disabled = false;
        }
    }
});

// Gérer le clic sur le bouton d'upload
uploadButton.addEventListener("click", function(event) {
    event.preventDefault();
    document.getElementById("upload-form").submit();
});
```

Figure 22 : Capture d'écran du code pour le "dropzone"

5. Lorsqu'un ou plusieurs fichiers sont déposés dans la zone de dépôt, l'événement "drop" est déclenché. Les fichiers sont récupérés à partir de l'objet "event.dataTransfer.files" et chaque fichier est vérifié s'il correspond à une image. Si c'est le cas, le nom du fichier est affiché dans l'élément "filename", les fichiers sont assignés à l'élément "input".

8. Plan de test et tests

8.1 Périmètre des tests

J'ai effectué mes tests sur Google Chrome.

Pour m'assurer que toutes les fonctionnalités de mon projet fonctionnent correctement, j'ai utilisé deux types de comptes : des comptes utilisateur en tant que joueur/staff et un compte entraîneur. Chaque jour, j'ai exécuté une série de tests pour vérifier que l'ajout d'une nouvelle fonctionnalité n'affecterait pas le fonctionnement d'une autre.

8.2 Équipement de test

Environnement

Durant le développement du projet, j'ai effectué mes tests sur l'environnement de développement suivant : Google chrome version 112.0.5615.138 (Build officiel) (64 bits) sous Windows version 10.0.19045 Build 19045.

Concernant le serveur, j'ai utilisé Apache version 2.4.41 Ubuntu version 20.04 avec PHP version 8.1.2, ainsi qu'un serveur MariaDB pour ma base de données.

Fichiers de test

Pour les tests, j'ai utilisé un compte entraîneur et plusieurs comptes utilisateur standard, et j'ai créé manuellement les événements. Toutes ces informations sont stockées dans la base de données et sont accessibles via le script SQL disponible dans le dossier /sql du projet.

8.3 Scénarios de test

Un scénario de tests pour évaluer les fonctionnalités principales de l'application est disponible ci-dessous. Ce scénario de tests vient en complément des tests unitaires, qui eux permettent de vérifier que chaque méthode fonctionne correctement. Le scénario de tests prévoit d'évaluer toutes les fonctionnalités disponibles dans le programme afin de s'assurer qu'elles fonctionnent toutes correctement.

Nom	1.1 Création d'un compte entraîneur
User story	S7 : Inscription pour l'entraîneur
Situation	Lorsqu'un visiteur arrive sur le site et souhaite se créer un compte entraîneur, il peut s'inscrire. En appuyant sur le bouton s'inscrire, si le compte est déjà créé il y a une erreur qui s'affiche. Si des informations sont fausses, un message s'affiche en fonction de l'erreur effectuée.
Résultats obtenus	Si le compte est existant ou que des champs sont faussement remplis, un message d'erreur s'affiche. Sinon, il est redirigé sur la page d'accueil avec la possibilité de créer une équipe dans un championnat qui existe déjà ou alors de créer un championnat puis une équipe.
Statut	OK

Nom	2.1 Création de championnats
User story	S8 : Gestion de championnats
Situation	Lorsqu'un entraîneur est identifié, il peut choisir de créer un championnat pour pouvoir y placer son équipe. Il doit entrer le nom du championnat, si ce champ est mal renseigné, un message d'erreur s'affiche.
Résultats obtenus	Un message d'erreur s'affiche si le champ n'est pas bien renseigné sinon le championnat est créé dans la base de données et l'entraîneur est redirigé sur la page d'accueil.
Statut	OK

Nom	2.2 Modifications / suppression championnats
User story	S8 : Gestion de championnats
Situation	Lorsqu'un entraîneur est identifié il peut supprimer et modifier les championnats qu'il a créés uniquement. Pour chaque action un message de confirmation s'affiche.
Résultats obtenus	Les messages s'affichent pour toutes les actions et les changements sont faits dans la BD.
Statut	OK

Nom	3.1 Création d'équipe
User story	S9 : Gestion d'équipe
Situation	L'entraîneur crée une équipe en rentrant les champs demandés. Des messages d'erreur s'affichent si des champs sont mal renseignés.
Résultats obtenus	Les messages d'erreurs s'affichent en cas d'erreur sinon il est redirigé vers la page pour créer les comptes des différents sportifs qu'il y aurait dans son équipe.
Statut	OK

Nom	3.2 Modifications / suppression d'équipe
User story	S9 : Gestion d'équipe
Situation	L'entraîneur peut supprimer ou modifier une équipe. Des messages de confirmations s'affichent pour chaque action.
Résultats obtenus	Les messages s'affichent pour toutes les actions et les changements sont faits dans la BD.
Statut	OK

Nom	4.1 Création de comptes joueur par l'entraîneur
User story	S10 : Gestion de sportif
Situation	L'entraîneur peut accéder à la page de création de comptes de sportif. Il peut créer un compte pour un staff ou joueur en entrant son email et un mot de passe.
Résultats obtenus	Un message d'erreur s'affiche si le compte est déjà créé ou que les champs sont mal renseignés. Sinon, le compte est bien créé et ajouté à la base de données.
Statut	OK

Nom	4.2 Modifications / suppressions sportif
User story	S10 : Gestion de sportif
Situation	L'entraîneur peut supprimer ou modifier un sportif. Des messages de confirmations s'affichent pour chaque action.
Résultats obtenus	Redirections sur la page équipe. Sinon message d'erreur sur la page de modification.
Statut	OK

Nom	5.1 Connexion
User story	S11 : Connexion
Situation	Un sportif possédant un compte peut se connecter en entrant son email et mot de passe. Un message d'erreur s'affiche si l'email ou le mot de passe est faux.
Résultats obtenus	Un message d'erreur s'affiche si les champs sont mal renseignés. Sinon, le sportif est connecté et redirigé sur la page d'accueil.
Statut	OK

Nom	6.1 Déconnexion
User story	S12 : Déconnexion
Situation	Un sportif connecté peut se déconnecter en appuyant sur le bouton « se déconnecter » dans la barre de navigation. Un message de confirmation s'affiche.
Résultats obtenus	Le message de confirmation s'affiche et s'il est validé, le sportif est déconnecté sinon il retourne sur la page d'accueil.
Statut	OK

Nom	7.1 Modifications du profil
User story	S13 : Modification du profil
Situation	Un sportif connecté peut modifier les informations sur son profil en se rendant dans la page profil. Il peut modifier sa photo de profil, son numéro, son nom etc. En appuyant sur confirmer un message de confirmation s'affiche.
Résultats obtenus	Le message de confirmation s'affiche et s'il est validé, les modifications seront effectuées dans la BD et le sportif reste sur la page profil avec un message comme quoi tout est bien modifié.
Statut	OK

Nom	8.1 Création d'un événements
User story	S14 : Gestion des événements
Situation	L'entraîneur peut créer un événement avec un nom une date un lieux et une description. À la création de l'événement un message qui montre que l'événement a bien été créé s'affiche.
Résultats obtenus	Le message de confirmations s'affiche et l'événement est créé dans la BD.
Statut	OK

Nom	8.2 Modifications / suppression d'événement
User story	S14 : Gestion des événements
Situation	Lorsqu'un entraîneur est identifié il peut supprimer et modifier les événements qu'il a créés uniquement. Pour chaque action un message de confirmation s'affiche.
Résultats obtenus	Les messages s'affichent pour toutes les actions et les changements sont faits dans la BD.
Statut	OK

Nom	9.1 Affichage des événements
User story	S15 : Affichage des événements
Situation	Quand un sportif est identifié, il peut voir tous les événements dans la page agenda sur le calendrier et les 5 prochains événements sur la page d'accueil.
Résultats obtenus	Les événements sont bien affichés sur ces deux pages.
Statut	OK

Nom	10.1 Invitations aux événements
User story	S16 : Invitations aux événements
Situation	L'entraîneur peut inviter les joueurs et staffs qu'il souhaite en les cochant dans une liste comportant tous les sportifs de l'équipe. Une fois l'invitation envoyée, les sportifs invités verront deux boutons sur l'événement pour choisir d'être présents ou absents.
Résultats obtenus	Les sportifs invités voient bien les deux boutons sur l'événement.
Statut	OK

Nom	11.1 Confirmations de présence
User story	S17 : Confirmations de présence
Situation	Les sportifs invités à l'événement peuvent confirmer leur présence en appuyant sur le bouton « Présent ». Si le sportif veut changer de choix, il peut toujours changer sa présence en cliquant sur l'événement et le changer dans la page de l'événement en question.
Résultats obtenus	Si le bouton « Présent » est cliqué, le bouton devient vert et le bouton « Absent » disparaît et s'il est recliquer, le bouton « Présent » disparaît et laisse place au bouton « Absence ».
Statut	OK

Nom	12.1 Confirmations d'absence avec un commentaire
User story	S18 : Confirmations d'absence avec un commentaire
Situation	Les sportifs invités à l'événement peuvent confirmer leur absence en appuyant sur le bouton « Absent ». Si le sportif veut changer de choix, il peut toujours changer sa présence en cliquant sur l'événement et le changer dans la page de l'événement en question.
Résultats obtenus	Si le bouton « Absent » est cliqué, un textbox apparaît pour pouvoir laisser une explication à cette absence et après confirmation, le bouton absent devient rouge et le bouton « Présent » disparaît.
Statut	OK

Nom	13.1 Visualisation des présences et commentaire
User story	S19 : Visualisation des présences et commentaire
Situation	L'entraîneur peut se rendre sur la page d'un événement et voir qui sera présent ou absent à l'événement et voir les excuses pour les absences
Résultats obtenus	Accès à la page de l'événement en cliquant sur l'événement en question et la liste des présences s'affiche.
Statut	OK

Nom	14.1 Gestion des scores pour les matchs terminés
User story	S20 : Gestion des scores pour les matchs terminés
Situation	En se rendant sur la page du match quand il est terminé, l'entraîneur peut ajouter un score, le modifier ou l'enlever.
Résultats obtenus	Accès à la page de l'événement en cliquant sur l'événement en question et après avoir validé le résultat tout l'effectif de l'équipe pourra consulter le score.
Statut	OK

Nom	15.1 Modification du rôle d'un sportif
User story	S21 : Modification du rôle d'un sportif
Situation	En se rendant sur la page gestion d'équipe, l'entraîneur peut changer les rôles des sportifs de l'équipe. Il doit confirmer les modifications pour qu'elle s'effectue.
Résultats obtenus	Accès à la page de gestion d'équipe que pour l'entraîneur. Un message de confirmation s'affiche quand il fait un changement de rôle puis et rediriger sur la page gestion d'équipe.
Statut	OK

Nom	16.1 Affichage des informations de l'équipe
User story	S22 : Affichage des informations de l'équipe
Situation	En se rendant sur la page « Mon équipe », chaque sportif peut voir les informations relatives à leur équipe.
Résultats obtenus	Accès à la page « Mon équipe » pour chaque sportif identifié. Une fois sur la page, toutes les infos de l'équipe s'affichent.
Statut	OK

Nom	17.1 Affichage des membres de l'équipe
User story	S23 : Affichage des membres de l'équipe
Situation	En se rendant sur la page « Mon équipe » et « accueil » chaque sportif peut voir les membres de leur équipe.
Résultats obtenus	Accès à la page « Mon équipe » pour chaque sportif identifié. Une fois sur la page, toutes les membres de l'équipe s'affichent. Pareil sur la page accueil.
Statut	OK

Nom	18.1 Affichage des informations du championnat
User story	S24 : Affichage des informations du championnat
Situation	En se rendant sur la page « Championnat », chaque sportif peut voir les informations relatives à tous les championnats.
Résultats obtenus	Accès à la page « Championnat » pour chaque sportif identifié. Une fois sur la page, tous les championnats s'affichent. En cliquant sur un championnat, on accède à ses informations.
Statut	OK

8.4 Évolution des tests

Num test	J1 Jeu, 27.04	J2 Mar 02.05	J3 Mer 03.05	J4 Jeu 04.05	J5 Lun 08.05	J6 Mar 09.05	J7 Mer 10.05	J8 Jeu 11.05	J9 Lun 15.05	J10 Mar 16.05	J11 Mer 17.05
1.1	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2.1	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
2.2	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
3.1	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
3.2	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
4.1	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
4.2	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓
5.1	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
6.1	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
7.1	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓
8.1	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓
8.2	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓
9.1	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓
10.1	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓
11.1	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓
12.1	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓
13.1	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓
14.1	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓
15.1	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓
16.1	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
17.1	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
18.1	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓

8.5 Échec des tests et raisons

Nom	2.1 Création de championnats
User story	S8 : Gestion de championnats
Situation	Lorsqu'un entraîneur est identifié, il peut choisir de créer un championnat pour pouvoir y placer son équipe. Il doit entrer le nom du championnat, si ce champ est mal renseigné, un message d'erreur s'affiche.
Raison de l'échec	La redirection en fonction de si l'entraîneur à une équipe ou non n'était pas encore implémenté.
Date de l'échec	Le 02.05

Nom	3.2 Modifications / suppression d'équipe
User story	S9 : Gestion d'équipe
Situation	L'entraîneur peut supprimer ou modifier une équipe. Des messages de confirmations s'affichent pour chaque action.
Raison de l'échec	La modification n'était pas encore implémentée.
Date de l'échec	Le 03.05

Nom	4.2 Modifications / suppressions sportif
User story	S10 : Gestion de sportif
Situation	L'entraîneur peut supprimer ou modifier un sportif. Des messages de confirmations s'affichent pour chaque action.
Raison de l'échec	Lorsque j'essayais de supprimer un entraîneur, une erreur se produisait car l'ID de l'entraîneur était référencé dans les tables "equipe" et "championnat". J'ai donc dû mettre en place une suppression en cascade pour résoudre ce problème.
Date de l'échec	Le 03.05 et 04.05

Nom	10.1 Invitations aux événements
User story	S16 : Invitations aux événements
Situation	L'entraîneur peut inviter les joueurs et staffs qu'il souhaite en les cochant dans une liste comportant tous les sportifs de l'équipe. Une fois l'invitation envoyée, les sportifs invités verront deux boutons sur l'événement pour choisir d'être présents ou absents.
Raison de l'échec	Sur la page d'invitation des sportifs, il était possible d'inviter tous les sportifs de l'application, et non seulement ceux de notre équipe.
Date de l'échec	Le 09.05

Nom	13.1 Visualisation des présences et commentaire
User story	S19 : Visualisation des présences et commentaire
Situation	L'entraîneur peut se rendre sur la page d'un événement et voir qui sera présent ou absent à l'événement et voir les excuses pour les absences
Raison de l'échec	Que ce soit en cas de présence ou d'absence, une croix s'affichait car les tests renvoyaient la même chose lorsque la valeur de présence était nulle ou égale à zéro.
Date de l'échec	Le 10.05

9. Conclusion

9.1 Difficultés rencontrées

En ce qui concerne les difficultés, je n'en ai pas eu énormément et elles étaient le plus souvent faciles à résoudre. Cependant je vais quand même citer certaines difficultés qui sont survenues pendant mon travail :

- La suppression d'équipe ou d'entraîneur.
Dans le cahier des charges il était écrit qu'aucune suppression devait se faire en cascade dans la base de données. Pendant la réalisation de mon site, j'ai dû apporter une modification à ce point après en avoir parlé à Mme Terrier et ensuite aux experts car la suppression d'équipe et d'entraîneur devait forcément amener la suppression des sportifs et des événements de l'équipe car sans entraîneur et équipe, les sportifs ne peuvent pas avoir de compte. Après la suppression d'une équipe, les événements liés à celle-ci ne servent plus.
- La différenciation entre un joueur présent, absent ou en attente de réponse.
Pour différencier les joueurs présents, absents ou en attente de réponse lors de l'affichage de chaque événement, j'ai rencontré un problème lors des tests. Je devais vérifier si la valeur de la variable "présent" était égale à 1, 0 ou null. Cependant, lors des tests, j'ai remarqué que le résultat était le même lorsque "présent" était égal à 0 ou null.
Pour résoudre ce problème, j'ai trouvé une solution en utilisant l'opérateur "===" au lieu de "==" dans les tests conditionnels. En utilisant trois signes égaux, le langage PHP compare d'abord le type de la valeur, puis la valeur elle-même. Étant donné que 0 et null n'ont pas le même type, j'ai pu différencier ces deux cas lors des tests. Ainsi, j'ai pu identifier si un joueur était en attente de réponse ou absent.

9.2 Planification

En ce qui concerne la planification, au début, j'étais dans les délais, ni en avance ni en retard, mais j'ai rapidement commencé à prendre de l'avance sur le développement du code dès le 3ème jour. En ce qui concerne les tâches initiales, j'ai dû modifier le nom de certaines d'entre elles car elles ne couvraient pas toutes les fonctionnalités que je devais implémenter. C'est lorsque j'ai abordé les tâches liées aux événements que j'ai vraiment pris de l'avance en consacrant un peu moins de temps à la documentation.

Je suis particulièrement satisfait de la planification réelle car le fait d'avoir eu 3 jours entiers pour la documentation m'a permis d'être bien organisé et efficace dans la rédaction de la documentation. Cela m'a permis de maintenir un bon équilibre entre le développement du code et la documentation, tout en veillant à ce que les deux aspects soient traités de manière approfondie et professionnelle et d'avoir du temps pour tester et régler les problèmes.

9.3 Améliorations possibles

Classement

Cette amélioration permettrait aux utilisateurs de suivre et de comparer les performances des différentes équipes participant aux championnats.

- **Système de points :**
On pourrait mettre en place un système de points où les équipes accumulent des points en fonction de leurs performances lors des matchs du championnat. Les victoires, les matchs nuls et les défaites pourraient être pris en compte pour déterminer les positions dans le classement.
- **Classement en temps réel :**
Afficher un classement en temps réel qui se met à jour automatiquement à mesure que les résultats des matchs sont enregistrés. Cela donnerait aux utilisateurs une vision actuelle et dynamique de la compétition.
- **Statistiques et détails des matchs :**
Fournir des statistiques détaillées pour chaque équipe, telles que le nombre de victoires, de défaites, de matchs nuls, ainsi que les buts marqués et encaissés. Les utilisateurs pourraient également consulter les détails des matchs pour obtenir des informations spécifiques sur les performances individuelles des équipes.

Envoie de lien d'invitation et notifications par e-mail

Cette fonctionnalité permettrait aux entraîneurs de facilement inviter les joueurs à rejoindre leur équipe et à participer aux événements.

- **Génération de lien d'invitation :**
Une fois que l'adresse e-mail du sportif est saisie, un lien d'invitation unique pourrait être généré pour chaque sportif. Ce lien servirait de moyen d'accès privilégié pour les joueurs invités à rejoindre l'équipe.
- **Envoi d'e-mail :**
Une fonction d'envoi d'e-mail automatisée qui enverrait un message personnalisé aux sportifs invités, contenant le lien d'invitation pourrait être incorporée.
- **Rappels d'invitation :**
Ajouter des rappels d'invitation pour les sportifs invités aux événements. Ces rappels consisteraient à l'envoi régulier d'e-mails à une fréquence prédéfinie, afin de permettre aux joueurs de confirmer leur présence aux différents événements.

9.4 Bilan personnel

La réalisation de ce projet a été une expérience extrêmement enrichissante pour moi. Drafteam s'est avéré être un projet formidable pour mettre en pratique mes compétences tout en traitant un sujet qui me passionne particulièrement : le football. J'ai vraiment apprécié l'ambiance pendant toute la durée du TPI, le fait de travailler sur un projet de A à Z, de la conception à l'évaluation.

Après avoir achevé l'intégralité de mon TPI, j'ai pris conscience de l'importance d'une bonne planification, de la mise en place de scénarios de tests solides, ainsi que de la discipline nécessaire pour éviter tout retard. Cela m'a permis de gagner en rigueur et en organisation, des compétences qui me seront précieuses dans mes projets futurs.

Je suis vraiment satisfait d'avoir pu concrétiser ce projet et de voir comment mes compétences acquises pendant mes années au CFPT sont complètes. En fin de compte, mon site répond pleinement à mes attentes. Je le trouve à la fois esthétique, fonctionnel et doté de fonctionnalités très intéressantes.

10. Remerciements

Je tiens à remercier :

- Mme Terrier pour son accompagnement et soutiens tout au long de mon TPI.
- Ma famille pour le soutien moral et l'encouragement.
- Mes camarades proches pour le soutien et la motivation.

11. Bibliographie

Code repris

- Template du site repris et modifié de mon ancien projet : <https://github.com/gotham226/CloseLake.git>
Lui-même repris de ce Template : [GitHub - dzarrorn/NFTwebsite html-sass](#)
- Peece Simple Router : <https://github.com/skipperbent/simple-php-router>

Sites utilisés

- Documentation PHP : <https://www.php.net/>
- Documentation FullCalendar : <https://fullcalendar.io/docs>
- W3Schools: [W3Schools Online Web Tutorials](#)
- DEMIVOLEE: <https://www.demivolee.com/creez-votre-compo/outil-de-creation-de-composition-dequipe/>

12. Table des illustrations

Figure 1 : Image qui représente les méthodologies	6
Figure 2 : Exemple du Trello du 2 mai	15
Figure 3 : Planning prévisionnel	16
Figure 4 : Planning réel.....	17
Figure 5 : Logo Git.....	18
Figure 6 : Logo Visual Studio Code	18
Figure 7 : Logo Linux avec logo Windows	18
Figure 8 : Logo FullCalendar	19
Figure 9 : Code pour la création d'un calendrier	19
Figure 10 : Logo JQuery	20
Figure 11 : Logo Composer	20
Figure 12 : Code montrant la création d'une route pour la page accueil	21
Figure 13 : Code montrant l'utilisation d'un namespace et d'une classe	21
Figure 14 : Image de l'arborescence	23
Figure 15 : Base de données du cahier des charges	24
Figure 16 : Base de données final	24
Figure 17 : Requête SQL pour l'inscription d'un sportif.....	25
Figure 18 : Requête SQL pour l'affichage des équipes participant à un championnat.....	25
Figure 19 : Requête qui renvoie le championnat actif s'il y en a un	26
Figure 20 : Capture d'écran du formulaire d'ajout de joueurs	26
Figure 21 : Capture d'écran de la page d'un événement	27

13. Annexes

Code Source : [Code.pdf](#)

Résumé TPI : [TPI Resume-GabrielMartin.pdf](#)

Documentation utilisateur : [TPI_DocumentationUtilisateur-GabrielMartin.pdf](#)