

Dossier de Projet



DMFC GAME

Site de paris sur les matchs de basket NBA

Présenté par

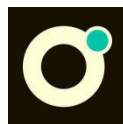
Charlotte Russell

Projet réalisé dans le cadre de la présentation au

Titre professionnel Développeur Web et Web Mobile

Centre de formation

École O'clock



Promotion

Xenon - 2023

Sommaire

Remerciements.....	4
Introduction.....	4
Résumé de Projet.....	5

Compétences couvertes par le projet

Développer la partie front-end.....	6
Développer la partie back-end.....	6

Organisation Du Projet

Équipe et rôles.....	7
Méthode agile.....	8

Cahier des charges

Introduction.....	9
Présentation de le projet.....	9
Objectifs de le projet.....	9
Fonctionnalités MVP.....	9 + 10
Versions futures.....	10
Arborescence.....	11
User-Stories.....	12
Routes.....	13 + 14

Graphique

WireFrames.....	15
Figma.....	16

Spécifications techniques

Technologies front-end.....	17
Technologies back-end.....	17
MCD.....	18
Sécurité web.....	19
Outils Utilisés.....	20 + 21

Réalisation

Réalisation Personnelle	21 - 27
Difficultés personnelles	28
Difficulté rencontrées en groupe	28
Où j'aurais pu être meilleure	29

Jeu d'essai

Données en entrée.....	29
Données attendues.....	30
Données obtenues.....	30

<u>Veille technologique sur la sécurité</u>	31
Sécurisation de DMFC.....	31

Sites anglophones

Utilisation de sites anglophones.....	32
Recherche sur un site anglais.....	32
Extrait d'un site anglophone.....	32
Traduction.....	33

<u>Conclusion</u>	34
L'avenir.....	34

Annexes

Wireframes.....	35 + 37
Header.....	38
Mettre à jour le profil	39
Infos NBA.....	40 + 41

Remerciements

Je tiens d'abord à exprimer ma gratitude à notre tuteur pédagogique Pierre-Yves dont le soutien a été essentiel tout au long de la formation. Je suis sûr que ce n'était pas un travail facile, mais sa présence pour répondre rapidement à nos questions et nous apporter son soutien était toujours là.

Merci à nos tuteurs, surtout à Gerard et Yannick. Leur expertise et leurs compétences pédagogiques ont vraiment simplifié une formation parfois complexe.

Merci à mes collègues de promotion. L'entraide était essentielle, et les rires partagés tout au long du parcours étaient une véritable source de motivation.

Merci à la région Bretagne qui a financé une grande partie de ma formation.

Enfin, un grand merci à l'équipe DMFC. Travailler en groupe sur un projet m'inquiétait, mais l'ambiance merveilleuse et les rires rendaient les choses faciles.

Introduction

Après avoir déménagé en France en 2018, j'ai travaillé dans des usines pour apprendre le français, une fois mon français atteint un niveau acceptable, j'ai décidé de reprendre mon métier de technicienne gazière. J'ai vite compris que cela ne me convenait plus. Non seulement le métier était complètement différent en France, mais le travail dur et pénible et les hivers froids dehors ne me plaisaient plus.

Je savais que j'avais besoin d'un travail qui répondrait à mon besoin de résoudre des problèmes, mais je ne savais pas par où commencer. J'ai suivi un cours court avec cob formation pour comprendre mes besoins en matière de travail et quel métier me conviendrait le mieux.

Développeur Web était le métier qui correspondait le plus à ma personnalité et qui m'avait déjà intéressé dans le passé. Pendant Covid, j'ai lancé une petite entreprise créant des sites web ciblant des mots-clés, puis vendant le travail à des artisans avec peu ou pas de présence en ligne, j'avais donc déjà un peu d'expérience. J'ai décidé que c'était la meilleure voie pour moi, et donc écrit à la région Bretagne pour demander du financement et mobilisé mon CPF pour payer le reste.

Après la fin de la formation de 6 mois, apprenant HTML, CSS, PHP, JavaScript et React, je peux dire avec plaisir que je ne regrette pas ma décision et que j'ai trouvé une carrière qui me plaira toute la vie.

Résumé de Projet

DMFC-Game. Un site de paris, juste pour le fun, pour les pronostics des matchs de basket NBA.

Le projet a été réalisé en utilisant les technologies ReactJS et Symfony. Nous étions une équipe de six développeurs, trois développeurs back-end et trois développeurs front-end. Suivant une méthodologie agile, nous avons eu une période de développement d'un mois, découpée en quatre étapes de 'sprints' successifs.

Le projet DMFC Game a pris forme pendant plusieurs années. À la base, il s'agissait d'une passion partagée par le product owner et ses amis : pronostiquer les saisons NBA. Ce hobby, auparavant géré via WhatsApp et un fichier Excel, s'est transformé en un véritable site internet lors de l'apothéose de notre formation.

Notre objectif était de créer un site permettant non seulement aux utilisateurs de placer facilement leurs paris, et voir leurs statistiques au classement de la ligue, mais également au DMFC (maître de jeu) de gérer facilement leur ligue. Le DMFC devait pouvoir créer une ligue, accepter et accéder aux joueurs de la ligue, créer des rondes et des matchs pour pronostiquer. Ainsi que consulter les pronostiques et statistiques des joueurs. Les joueurs devaient pouvoir voir et parier sur la liste des matchs sélectionnés par la DMFC, consulter leurs statistiques et résultats et consulter le classement général.

En l'espace d'un mois, nous avons créé un site web épuré, efficace et élégant, désormais accessible en ligne sur www.dmfc-game.fr.

Compétences Couvertes Par le Projet

Développer la partie front-end

- **Maquetter une application:** Nous avons d'abord complété les user-stories et l'arborescence pour avoir une meilleure compréhension de ce qui était requis pour le site. Nous avons ensuite créé les wireframes, en commençant par mobile car ce serait l'appareil le plus utilisé pour la site. Nous avons travaillé en équipe lors de la création des wireframes. Nous avons utilisé whimsical et communiquant via discord.
- **Réaliser une interface utilisateur web statique et adaptable:** Le site est entièrement responsive sur mobile, tablette et desktop. Nous avons opté pour une approche mobile-first pour garantir que le site soit adaptable. En utilisant cette stratégie, nous avons assuré que notre interface est fonctionnelle et orientable, quelle que soit la plateforme utilisée.
- **Développer une interface utilisateur web dynamique:** Nous avons choisi d'utiliser ReactJs pour développer notre site. React est basée sur des composants plutôt que des pages et cela donne une grande dynamique à notre site, car il s'agit d'une seule page. L'expérience est donc beaucoup plus fluide puisqu'il y a très peu de rechargements de page.

Développer la partie back-end

- **Créer une base de données:** La conception de notre base de données a impliqué la création de 9 tables distinctes, un processus réalisé en utilisant Symfony et la librairie Doctrine. Nous avons créé cela en suivant le modèle conceptuel des données (MCD) que nous avons déjà élaboré.
- **Développer les composants d'accès aux données:** La base de données ont été construits avec Symfony et la bibliothèque Doctrine. Les composants servent de pont, entre le front-end et back-end. Ces composants se manifestent sous forme de modèles, nous avons défini la structure des données et les requêtes pour permettre au front-end d'interagir avec la base de données et de récupérer, créer et modifier des données.
- **Développer la partie back-end d'une application:** Notre back-end a été développé sous la forme d'une API (Interface de Programmation Applicative) pour permettre des requêtes et une communication facile depuis le front-end. En suivant la méthode CRUD (create, read, update, delete), nous avons déterminé et créé les endpoints nécessaires pour notre site.

Organisation de Projet

Équipe et Rôles

L'équipe était composée de trois développeurs front-end et trois développeurs back-end chacun avec son propre rôle.

- **Product Owner : Fabien Baraille.**
L'idée initiale du jeu DFMC est venue de Fabien et de ses amis. Il était donc clair qu'il serait le product owner. Il communiquait régulièrement avec le client pour s'assurer que la vision du produit était respectée, , validait les fonctionnalités et réévaluait les priorités.
- **Scrum Master : Maxime Lemarchand.**
En tant que scrum master, Maxime était responsable de l'organisation quotidienne du projet en suivant les principes scrum. Sa mission était de vérifier que l'équipe atteigne ses objectifs dans les délais impartis pour s'assurer du bon déroulement du projet et garantir une compréhension commune des objectifs et des tâches à accomplir, en utilisant trello.
- **Lead Développeur Front-end : Quentin Dumet.**
En tant que développeur front-end principal, Quentin était responsable de la gestion de la parti front-end. Il a supervisé l'architecture et la mise en œuvre des fonctionnalités frontales en s'assurant que les standards de qualité et les bonnes pratiques de codage soient respectées.
- **Lead Développeur Back-end et Technical Referent Back-end: Quentin Rivière.**
En tant que développeur back-end principal, Quentin était responsable de la gestion de la parti back-end. Il a supervisé la conception, le développement et la maintenance des fonctionnalités Symfony, en garantissant le respect des standards de qualité, des bonnes pratiques de développement et la sécurité de l'architecture logicielle. Ainsi que la recherche des packages Symfony supplémentaires.
- **Git Master : Tony Barbier.**
En tant que git master, Tony était en charge de superviser l'utilisation de Git, de gérer les branches, de résoudre les conflits et d'organiser le workflow, pour assurer une gestion efficace des versions dans le projet.
- **Technical Referent Front-end : Charlotte Russell.**
En tant que référent technique front-end, j'étais responsable de la recherche, de l'évaluation et de la recommandation de packages, bibliothèques et de ressources React supplémentaires. J'ai également recherché de nouvelles technologies et solutions pour améliorer l'expérience utilisateur.

Méthode Agiles

Nous avons suivi la méthodologie agile, avec une période de développement d'un mois, découpée en quatre étapes de 'sprints' successifs. La phase conception sprint 0-1 et la phase développement sprint 2-3.

Tous les jours, nous nous réunissions à 9 heures du matin pour faire un débriefing, puis nous restions soit en groupe sur Discord, soit nous nous séparions en 2 groupes, front-end et back-end. Cela a facilité la communication entre nous et nous a permis d'éviter que deux personnes travaillent sur la même pièce ou soient coincées trop longtemps sur des problèmes.

Sprint 0:

Pendant le sprint 0 nous avons débriefer et décider des rôles. Nous avons ensuite complété le cahier des charges. Pour cela, nous avons discuté du besoins et objectifs du projet, le fonctionnalités y compris de la MVP et contenu pour les versions futures. Vient ensuite la discussion sur les technologies. Ce que nous utiliserions pour la gestion des routes, les API, l'authentification, la création de back-office et gestion de mot de passe oublié. Ensuite nous avons construit l'arborescence de site, les routes front-end et back-end, les user-stories et les wireframes.

Sprint 1:

Pendant le sprint 1 nous avons revu le cahier des charges ensemble. Nous nous concentrons ensuite sur la création de l'infrastructure back-end. Avec le création de le contrôleur API pour gérer les ligues et les utilisateurs et le configuration des jetons JWT pour l'authentification. Nous avons créer le base de données et les premier routes côté back et les premier composants côté front-end.

Sprint 2:

Pendant le sprint 2 nous sommes entrés dans la phase de développement. Nous avons commencé par la création des fonctionnalités clés de notre produit minimum viable (MVP) pour le maître de jeu, le joueur et l'admin. Nous avons ensuite terminé plus ou moins les routes back-end.

Sprint 3:

Pendant le sprint 3 nous étions en plein mode développement. Nous avons finalise toutes les fonctionnalités. Nous avons résolu des bugs et fait l'ajustement global du CSS. Nous avons également eu le temps d'ajouter quelques fonctionnalités de version 2. Une liste de tâches dans Google Docs nous a aidés à prioriser et à marquer nos contributions et nous a permis de finaliser efficacement la finalité de le site.

Cahier des Charges

Un cahier des charges est crucial lors du lancement d'un projet. Il sert de guide le développement, permet de définir clairement les objectifs, contribue à une communication efficace entre les parties et constitue une base d'évaluation et de validation.

Présentation de le projet

Un concept en gestation depuis 10 ans, le client et ses amis jouaient des paris entre eux pour les saisons de la NBA. Ils ont pronostiqué les matchs de la saison régulière, les rounds, les classements de fin de saison et les playoffs.

Un maître de jeu a géré ces paris, choisissant les ligues et les matchs pour les paris, enregistrant les choix des joueurs et calculant les résultats tout au long de la saison. Il était géré par un groupe WhatsApp où les joueurs envoyaient leurs paris et le maître du jeu saisissait ensuite ces paris dans une feuille Excel.

Le projet cherche à simplifier ce jeu amical en développant une site qui facilite le rôle du maître de jeu tout en offrant une expérience de pari plus fluide aux joueurs.

Objectifs de le projet

De toute évidence, l'objectif principal du projet est de simplifier le travail du maître du jeu. Avec la gestion du classement de la ligue et le calcul automatiquement des points et statistiques.

Pour les joueurs, une expérience dynamique et fluide est également requise. La possibilité de pouvoir sauvegarder un pari et d'y revenir plus tard, voir en temps réel leur position dans le classement général et comparer leurs résultats avec ceux des autres dans la ligue est aussi un must absolu.

Fonctionnalités MVP

Ce sont les fonctionnalités MVP que nous avons identifiées comme essentielles pour le lancement efficace du projet. Ces fonctionnalités centrales permettent de répondre aux besoins fondamentaux de le maître de jeu et les joueurs.

Fonctionnalités pour le Maître du jeu :

- Créer sa ligue.
- Valider inscription d'un joueur.

- Création de round.
- Choix des matchs saison régulière (round par round.)
- Renseigner les résultats des matchs(round par round.)
- Automatiser les calculs de point des joueurs dès renseignement des résultats des matchs.
- Consulter les pronostics de chaque joueur.
- Consulter les résultats des joueurs et leurs statistiques.

Fonctionnalités pour le Joueur :

- Création de compte.
- Choix de la ligue.
- Pronostics (round par round.)
- Consulter ses résultats (round par round.)
- Consulter les résultats des autres joueurs de la ligue.
- Consulter le classement général de la ligue.

Fonctionnalités pour l'Admin :

- Gérer les ligues.

Versions futures

Nous avons envisagé les fonctionnalités suivantes comme futures fonctionnalités de la version:

Fonctionnalités de la version 2 :

- Pronostics du Top 10.
- Validation des Tops 10 à la fin de la saison.
- Trois niveaux de validation des pronostics (sauvegarde temporaire, validation, validation avec publication publique sur le canal de discussion.)

Fonctionnalités de la version 3 :

- Gestion des pronostics pour les playoffs (validation des équipes par le DMFC et pronostics des joueurs.)
- Espace de discussion libre (TrashTalk.)
- Intégration de WhatsApp / e-mail pour des rappels concernant les pronostics.

Fonctionnalités de la version 4 :

- Possibilité d'importer des données depuis des fichiers Excel existants.

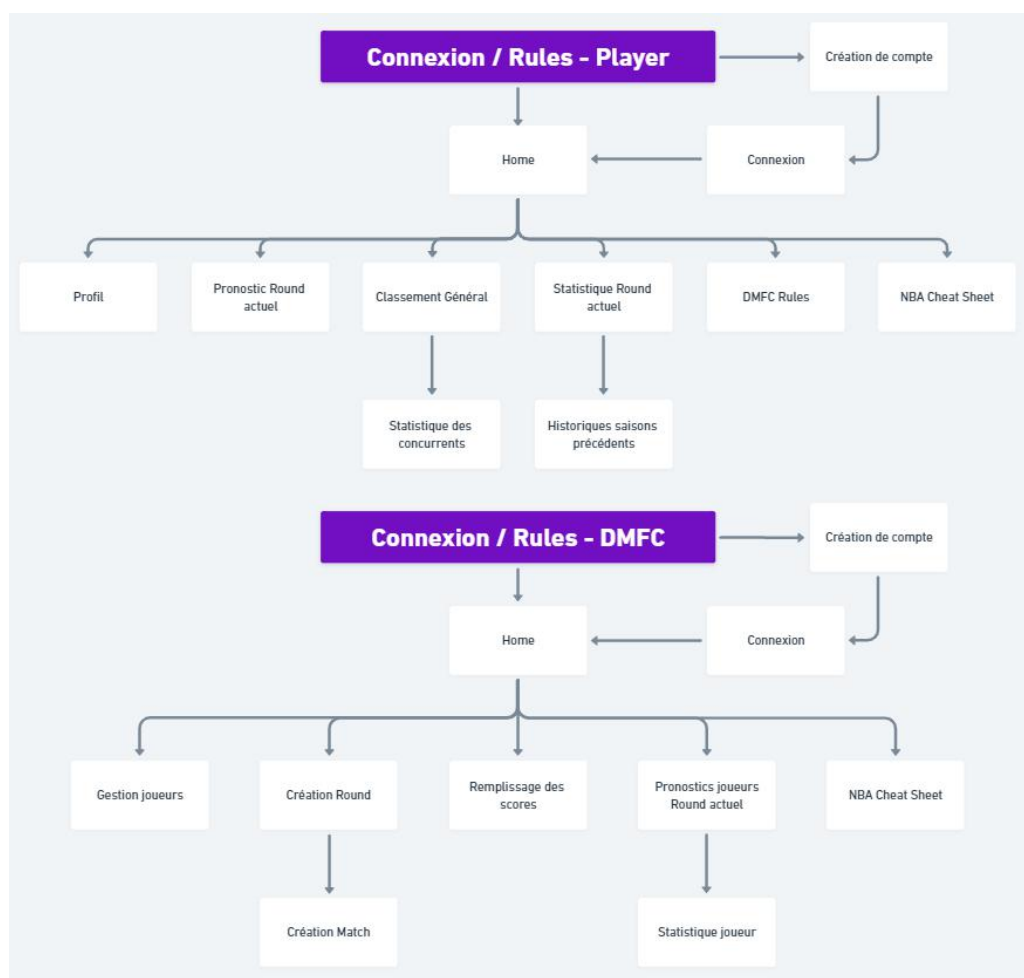
Arborescence

L'arborescence d'un site web est essentielle car elle montre la structure et l'organisation de toutes les pages et les liens entre chaque page. L'arborescence représente le squelette ou la structure du site et indique comment son contenu est regroupé, lié et présenté à l'utilisateur.

Nous avons créé deux arborescences distinctes:

L'arborescence du joueur: Chaque section est centrée sur les aspects individuels du jeu et la consultation des informations pertinentes pour les participants. Sauf les quatre pages en commun: Le création de compte, le connexion, le page d'accueil et le page de NBA cheatsheets.

L'arborescence du maître du jeu: Chaque section est centrée sur la gestion et la supervision du jeu. Sauf les quatre pages en commun: Le création de compte, le connexion, le page d'accueil et le page de NBA cheatsheets.



User-Stories

Les user-stories sont une partie très importante du développement de projets, elles décrivent, de manière concise mais détaillée, ce que les utilisateurs veulent accomplir avec le système. Ainsi que aident à prioriser les besoins des clients et à minimiser les risques potentiels. Ils assurent également la collaboration entre les équipes et organisent la liste des fonctionnalités à développer (product backlog) en fonction des priorités.

Nous avons créé nos user-stories lors du sprint 0. Cela nous a aidé à organiser, prioriser et déléguer efficacement le travail avant de démarrer le projet. Elles sont servi comme un pont entre l'équipe de développement et les besoins des utilisateurs.

USER STORIES

ADMIN

En tant que	Je veux	Afin de (si besoin/nécessaire)
Admin	Avoir accès aux utilisateurs et aux ligues	Valider les inscriptions et gérer les différentes ligues
Admin	Avoir accès aux équipes NBA	Pouvoir modifier les informations sur les équipes de NBA
Admin	Avoir accès aux saisons	Pouvoir créer et modifier les informations sur les saisons

DMFC - Maître du jeu

En tant que	Je veux	Afin de (si besoin/nécessaire)
DMFC	Créer mon compte et ma ligue	-
DMFC	Accès à la liste des joueurs de la ligue	Valider les inscriptions et gérer les joueurs actifs/inactifs
DMFC	Créer les rounds	Créer un round pour y affecter des matchs
DMFC	Créer les matchs à pronostiquer	Choisir les matchs sur lesquels parier et les deadlines
DMFC	Renseigner les résultats des matchs	Automatiser les calculs de points des joueurs
DMFC	Renseigner les cotes officielles	Comparer les pronos joueurs et cotes officielles
DMFC	Consulter les pronostiques des joueurs	-
DMFC	Consulter les statistiques des joueurs	D'alimenter la news letter

Joueur

En tant que	Je veux	Afin de (si besoin/nécessaire)
Joueur	Créer mon compte	Rejoindre une league et participer aux pronostics
Joueur	Voir la liste des matchs sélectionnés par le DMFC	Réaliser des pronostiques (gagnant et différence de points)
Joueur	Consulter ses stats et résultats	Garder un historique des pronostics réalisés
Joueur	Consulter le classement général	Pouvoir accéder aux stats et fiches des différents joueurs

Routes

Pendant le sprint 0, nous avons travaillé en équipe pour créer tous les routes nécessaires pour le front-end et pour le back-end. En utilisant l'arborescence et les user-stories comme référence, cela nous a aidés à comprendre la structure de notre site. Cette méthode a facilité la planification en nous permettant de visualiser de manière détaillée les trajets qui seraient nécessaires pour la navigation de notre site.

Front-end Routes

Routes

DMFC

En tant que	Route	Description
DMFC	/league-management	Route pour accéder à la page de gestion des joueurs de la ligue
DMFC	/creation/SR	Route pour accéder à la page permettant de créer les rounds et matchs de la saison régulière
DMFC	/scores/SR	Route pour accéder à la page permettant de remplir le score et les cotes des matchs

Joueurs

En tant que	Route	Description
Joueur	/profil	Route pour accéder à la page permettant l'édition de son profil
Joueur	/player-bet	Route pour accéder à la page permettant de pronostiquer les matchs du round actuel
Joueur	/roundsStat	Route pour accéder à la page permettant de voir les résultats round par round

Commun

Route	Description
/	Route pour accéder à la page d'accueil pour les utilisateurs connectés
/login	Route pour accéder à la page de connexion ou de création de compte
/logout	Route pour accéder à la page de déconnexion
/profil	Route pour accéder à la page permettant d'éditer son profile
/rankings	Route pour accéder à la page avec le classement général de la ligue
/player/playername	Route pour accéder à la page permettant de voir les statistiques d'un joueur de la ligue
/rules	Route pour accéder à la page contenant les règles et le déroulement du DMFC Game
/cheatsheets	Route pour accéder à la page recensant des liens utiles pour les pronostics
/terms-and-conditions	Route pour accéder à la page contenant les règles et le déroulement du DMFC Game

Back-end Routes

NEWS

ENDPOINT	Method	Description
/api/news	GET	Get all news
/api/league/{id}/news	GET	Get all news by a league id
/api/news/new	POST	Create a news
/api/news/{id}	PUT	Update a news
/api/news/{id}	DELETE	Delete a news

ROUNDS

ENDPOINT	Method	Description
/api/league/{id}/round	GET	Get all rounds for a league
/api/round/new	POST	Create a round
/api/round/{id}	PUT	Update a Round
/api/round/{id}	DELETE	Delete a Round

SEASONS

ENDPOINT	Method	Description
/api/seasons	GET	Get all seasons
/api/seasons/{id}	GET	Get a season by id
/api/seasons	POST	Create a season
/api/seasons/{id}	PUT	Update a season
/api/seasons/{id}	DELETE	Delete a season

SRPREDICTIONS

ENDPOINT	Method	Description
/api/srprediction/{id}	GET	Get all predictions by a user id
/api/game/{id}/srprediction	GET	Get all predictions by a game id
/api/srprediction/new	POST	Create a prediction
/api/prediction/update/{id}	PUT	Update a prediction
/api/prediction/{id}/dmfc	PUT	Update a prediction by a DMFC

TEAMS

ENDPOINT	Method	Description
/api/teams	GET	Get all NBA teams infos
/api/team	POST	Create a NBA team
/api/team/{id}	PUT	Update a NBA team infos
/api/team/{id}	DELETE	Delete a NBA team

USERS

ENDPOINT	Method	Description
/api/users	GET	Get all users infos
/api/user/{username}	GET	Get infos of a user by his username
/api/league/{id}/users	GET	Get all users infos by a league id
/api/user/new	POST	Create a user
/api/user/{id}	PUT	Update a user's infos
/api/user/{id}/dmfc	PUT	Update a user's infos by a DMFC
/api/user/{id}	DELETE	Delete a user

Graphiques

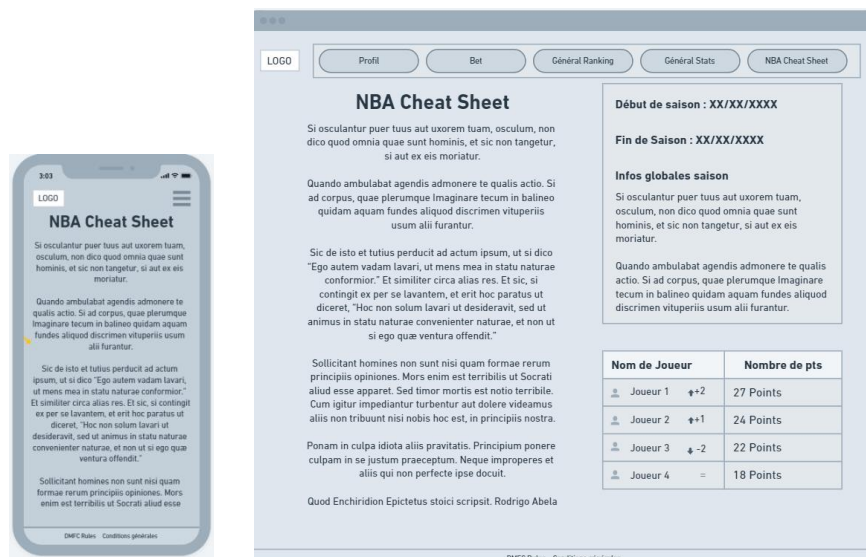
WireFrames

Au début d'un projet, une étape cruciale de la production est celle des wireframes. Ce sont des modèles simples, sans les détails graphiques qui donnent une représentation visuelle du site web. Ils permettent d'établir la structure initial du site avant de passer aux étapes de conception et de développement. Ils donnent également la possibilité au client d'apporter les modifications nécessaires dès le début.

Nos wireframes ont été construits en utilisant whimsical et nous avons travaillé en groupe pour les compléter. Nous avons discuté en groupe des fonctionnalités et du contenu que nous souhaitions sur chaque page permettant de définir les détails précis de chaque page avant de démarrer la phase de développement. Ces échanges collectifs ont amélioré la qualité de nos wireframes en apportant une vision partagée et détaillée des besoins et de la structure de l'interface.

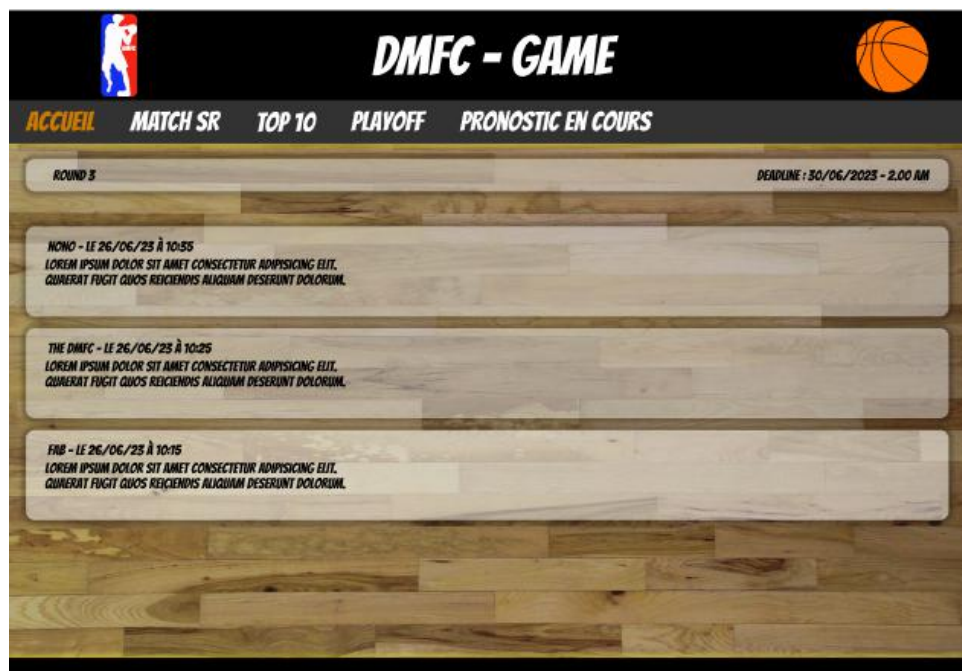
Voici un exemple d'un des wireframes, le reste se trouve dans l'annexe "Wireframes" en pages 35 à 37.

Version desktop et mobile de la page cheat sheets (NBA Info.)



Figma

Pour obtenir une vue plus complète de la conception, nous avons utilisé Figma pour créer une maquette détaillée de l'interface utilisateur. Cet outil de conception nous a permis de visualiser l'apparence du produit final. En développant des concepts visuels détaillés, nous, les développeurs front-end, disposons d'une référence précise pour les couleurs, les graphiques et la mise en page. Ce modèle a également servi de guide visuel au client, facilitant ainsi les discussions sur les modifications et ajustements nécessaires. Cette approche nous a permis de gagner du temps et de maintenir une communication claire tout au long du processus de développement.



Spécifications Techniques

Technologies front-end

Nous avons développé le front-end de cette projet avec ReactJS et également utilisé NPM (Node package manager) pour gérer quelques parties du contenu. De plus, nous avons intégré Vite.

Pour la gestion des requêtes API nous avons utilisé RedAxios. Redaxios est une alternative plus légère à Axios, offrant une API similaire mais avec une plus petite taille.

Pour la gestion des routes nous avons utilisé React-router-dom. React Router DOM est une bibliothèque essentielle pour la navigation dans les applications React.

Pour la gestion de store nous avons utilisé Redux. Redux facilite la gestion des données partagées entre différents composants de le site, en conservant un global state.

Pour la gestion de CSS nous avons utilisé Sass. Sass est une extension de la syntaxe CSS qui offrant des fonctionnalités supplémentaires telles que les variables.

Pour la gestions des messages d'erreur ou succès nous avons utilisé Toastify. Toastify est une bibliothèque qui permet de créer et de personnaliser des notifications de manière très flexible et dynamique.

Technologies back-end

Nous avons développé le back-end de ce projet avec Symfony.

Pour la création de le back-office nous avons utilisé Twig.

Pour la gestion de la base de données, nous avons utilisé Doctrine.

Pour la génération de données aléatoires, nous avons utilisé Faker.

Pour la gestion des jetons JWT nous avons utilisé Lexik JWT Authentication Bundle.

Pour la gestion des CORS nous avons utilisee Nelmio Cors.

Pour gérer un mot de passe oublie nous avons utilisé Reset Password Bundle.

MCD

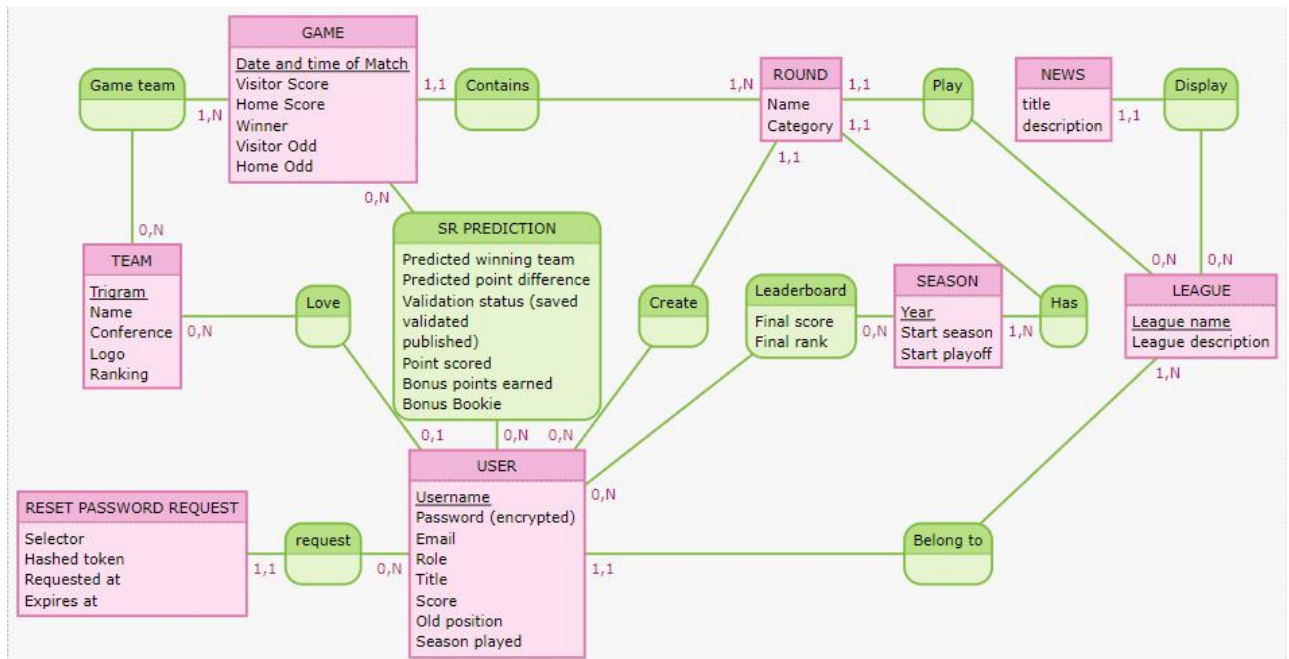
Notre processus de conception du Modèle Conceptuel de Données (MCD) a été une étape cruciale dans la construction de l'architecture de notre base de données. Lors de la création du MCD, nous avons visualisé les relations entre les différentes entités de notre système en utilisant l'outil Mocodo, ce qui nous a permis d'avoir une vue d'ensemble claire de la structure de notre base de données.

La structure de notre MCD a été une tâche un peu complexe, en raison du nombre des relations importantes à gérer. Nous avons identifié deux relations particulièrement critiques, celles entre les tables SR Prediction et Leaderboard. Ces relations many-to-many contiennent des informations supplémentaires et représentent des données cruciales pour notre système.

La table 'user' a aussi joué un rôle central dans notre MCD, avec de multiples relations provenant de cette entité centrale. Ces relations étendent leur influence à presque toutes les entités de la base de données.

La création de notre MCD a nécessité beaucoup de réflexion autour des relations et des entités et nous avons eu besoin d'un peu d'aide de la part des tuteurs pour le réaliser.

Voici la version finale de notre mcd:



Sécurité Web

React est déjà très sécurisé. Il offre une protection intégrée contre les attaques de Cross-Site-Scripting grâce à sa méthode de rendu d'éléments. Il encode automatiquement les données avant de les afficher, éliminant ainsi le risque d'ajout de scripts malveillants.

Le site est hébergé avec un certificat SSL, il utilise donc HTTPS pour crypter les données transmises entre le client et le serveur.

Nous avons utilisé les jetons JWT (JSON Web Tokens) pour sécuriser notre site. Ces jetons sont attribués aux utilisateurs lorsqu'ils se connectent et contiennent leurs informations d'identification et leurs autorisations. Chaque fois qu'une requête est faite le serveur peut vérifier que l'utilisateur est autorisé à accéder ou effectuer ce qu'il a demandé, pour garantir l'accès autorisé. C'est une méthode de sécurité solide car ça évite le besoin de stocker les informations d'authentification côté serveur parce que les informations sont contenues dans le jeton lui-même. De plus, les jetons sont signés, de manière sécurisée, pour prévenir toute altération.

Nous avons aussi sécurisé les routes côté front-end et back-end en définissant des rôles ce qui restreint l'accès à certaines requêtes et données spécifiques.

Outils Utilisés

Versioning

- **GitHub:** GitHub a été notre principal outil pendant le projet. Nous l'avons utilisé pour collaborer en équipe en partageant notre code, travailler sur des branches distinctes et pour résoudre les conflits entre les versions différentes. Cela nous a permis de versionner et coordonner notre code entre les six développeurs.

Développement

- **VSCode:** Visual Studio Code a été notre environnement de développement. C'est là que nous avons écrit, structuré et modifié notre code. Nous avons également utilisé quelques extensions pour aider pendant le développement, tels que Liveshare. Cette extension permet à un ou plusieurs développeurs de collaborer sur le même dossier en temps réel. Cela a facilité le processus de débogage.
- **Redux Dev Tools:** Redux Dev Tools a été un outil indispensable pour la gestion des erreurs. En temps réel, nous pouvions suivre les objets et vérifier s'ils avaient été correctement envoyés vers le state ou les actions.
- **Mocodo:** Nous avons utilisé Mocodo pour créer le modèle conceptuel de données (MCD.) Cela nous a permis de visualiser la structure de notre

base des données et représentante des entités, leurs attributs et des relations entre elles.

Gestion de projet

- **Trello:** Nous avons utilisé Trello pour gérer les tâches et organiser la méthode agile. Nous avons 3 tables. spring backlog, product backlog et doing. Chaque carte avec sa couleur appropriée pour aider à les catégoriser, les prioriser et les différencier.

Documentation

- **Google Drive:** Nous avons un dossier Google Drive où nous stockions tous les documents partagés entre l'équipe tels que le cahier des charges et la liste des tâches ainsi que les captures du mcd, des routes et de l'arborescence.

Communication

- **Discord:** Nous avons communiqué pendant toute la durée du projet via Discord. tous les matins à 9h, nous nous retrouvions dans un canal pour un petit débriefing et la plupart du temps nous passons la journée en deux groupes front-end et back-end. Cela a permis une communication maximale entre nous et a permis d'économiser beaucoup de temps et d'efforts. Le partage d'écran a également été d'une grande aide pour le débogage et la présentation du travail effectué pendant la journée afin de s'assurer que tout le monde était satisfait de la production. Nous avons également eu nos débriefings hebdomadaires avec nos tuteurs via Discord.
- **Slack:** Nous avons utilisé slack principalement pour l'entraide entre nos collègues de la promotion. Nous avons également échangé des messages avec les tuteurs et eu des rappels pour les réunions de groupe, comme les présentations, via cette plateforme.

Graphique

- **Whimsical:** Nous avons utilisé Whimsical pour la création des schémas wireframes, l'arborescence et les routes. Nous avons trouvé cet outil particulièrement utile en raison de sa vaste bibliothèque d'outils et de modèles. Nous pourrions également travailler ensemble sur les schémas facilitant la collaboration entre nous.
- **Figma:** Nous avons utilisé Figma pour créer une représentation visuelle de notre site web avant de commencer la conception. Cela a nous permette d'avoir une image en tête de la structure du site avec des couleurs et graphiques. Cela a facilité la compréhension entre moi et les autres développeurs front-end de visualiseur le site web et fluidifié la phase de conception. Ça a également donné

au client l'opportunité de visualiser et ajuster les détails esthétiques avant le début du développement.

Ressources

- **Google:** J'ai utilisé Google pour rechercher des bibliothèques et des technologies pour React. Nous avons également utilisé Google pour nous aider à déboguer avec des sites tels que www.stackoverflow.com.

Réalisation

Réalisation Personnelle

J'ai concentré tout mon temps pendant le projet sur le développement front-end. Les seules fois où j'ai regardé le back-end, c'était pour trouver les routes correctes pour les requêtes API dans le middleware.

J'ai travaillé principalement sur les pages "profil" et "NBA Infos", construisant moi-même la grande majorité du code et des styles des deux pages. J'ai également créé le header et le footer, ainsi que la barre de navigation et le menu déroulant. J'ai également travaillé sur d'autres petites fonctionnalités comme le package "Toastify" pour les messages de réussite et d'erreur.

Header

Le header était une tâche assez simple. J'ai regroupé les logos dans une balise "div" pour simplifier le processus de conception. J'ai ensuite utilisé "display: none" sur le deuxième logo, car les deux logos sur le header avec le menu déroulant étaient trop. Une fois que la taille de l'écran est supérieure à 768 pixels, "display : unset" remplace "display : none" et le deuxième logo apparaît. J'ai utilisé "display: flex" et "justify-content: space-evenly" pour positionner les icônes de chaque côté de le header. Vous pouvez retrouver des captures du code en détail sous l'annexe "Header" sur page 38.

J'ai utilisé la condition "isLoggedIn", au titre h1, pour transformer le texte, en lien vers la page d'accueil, uniquement lorsque l'utilisateur est connecté.

```
<h1 className='header-title'>{isLoggedIn ? <Link to="/">DMFC - Game</Link> : "DMFC - Game"}</h1>
```

Page de profil

J'ai travaillé principalement seul sur la page de profil. J'ai été chargé de créer une interface affichant l'e-mail, le nom d'utilisateur, le score, la ligue et l'équipe préférée des joueurs. De plus, j'ai implémenté un formulaire permettant de modifier l'e-mail, le mot de passe, l'identifiant et l'équipe préférée.

Afficher les données du profil

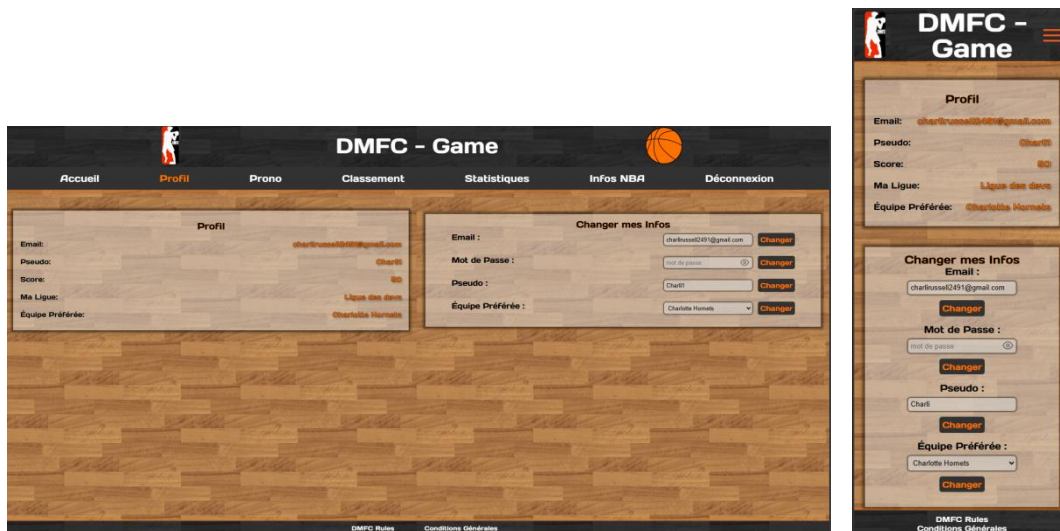
Pour accéder aux informations nécessaires à l'affichage des données, j'ai utilisé l'objet `loggedUser`, qui extrait les données du Redux store.

```
<div className="profil">
  <h3>Profil</h3>
  <p>Email: <span className="perso">{loggedUser.email}</span></p>
  <p>Pseudo: <span className="perso">{loggedUser.username}</span></p>
  <p>Score: <span className="perso">{loggedUser.score ? loggedUser.score : 0}</span></p>
  <p>Ma Ligue: <span className="perso">{loggedUser.league_id.leagueName}</span></p>
  <p>Équipe Préférée: <span className="perso">{loggedUser.team !== null ? loggedUser.team.name : 'Non choisi'}</span></p>
</div>
```

Mis à jour des données

Pour le formulaire de modification des données, j'ai mis en place les fonctions 'handleSubmit' et 'handleInput'. Lorsque l'utilisateur modifie un champ et appuie sur le bouton 'Changer', ces fonctions sont appelées. Les valeurs sont enregistrées dans le Redux store grâce à 'setInputValue' et 'dispatch'. Les données sont collectées dans 'updateUserData'. Ensuite, l'action 'UPDATE_USER_PROFILE' est déclenchée, appelant l'API pour mettre à jour les informations via une requête 'PUT' vers le serveur. Les extraits de code sont fournis dans l'annexe "Mettre à jour du profil" en page 39.

Page de profil version desktop et mobile.



Fléchés

Pour faciliter la navigation, j'ai mis en place des flèches de navigation pour faire défiler entre les joueurs du classement. J'ai réalisé cela avec le code ci-dessous:

```
const nextPlayer = () => {
  const nextIndex = (playerIndex + 1) % userDataList.length;
  setPlayerIndex(nextIndex);
};

const previousPlayer = () => {
  const previousIndex = (playerIndex - 1 + userDataList.length) % userDataList.length;
  setPlayerIndex(previousIndex);
};
```

Ces fonctions naviguer vers les joueurs suivants et précédents dans la “userDataList” utilisent le “playerIndex” actuel pour déterminer le prochain ou le précédent utilisateur dans la liste et mettent à jour l'état “playerIndex”, permettant ainsi de suivre quel utilisateur est affiché à tout moment. Ensuite j'ai utilisé le composant “Link” autour d'un bouton pour pouvoir naviguer entre joueurs. Comme on le voit dans le code ci-dessous:

```
<Link to={` /player/${userDataList[nextPlayerIndex].username}`}>
  <button className='arrow' onClick={nextPlayer}>&gt;</button>
</Link>

<Link to={` /player/${userDataList[previousPlayerIndex].username}`}>
  <button className='arrow' onClick={previousPlayer}>&lt;</button>
</Link>
```

Voici les boutons fléchés:



Bouton retour

```
import PropTypes from 'prop-types';
import { Link } from 'react-router-dom';
import './Retour.scss';

function Retour ({ where, link }) {
  return (
    <div>
      <button className="return-btn" type="button">
        <Link to={link}>{`Retour ${where}`}</Link>
      </button>
    </div>
  );
}

Retour.propTypes = {
  where: PropTypes.string.isRequired,
  link: PropTypes.string.isRequired,
};

export default Retour;
```

Pour faciliter la navigation des utilisateurs non connectés, j'ai ajouté un bouton de retour sur les pages accessibles à ces utilisateurs. J'ai créé un nouveau composant appelé “Retour” qui affiche un bouton redirigeant vers “where” lorsque l'utilisateur n'est pas connecté. Ce bouton utilise “Link” de pour la redirection.

les PropTypes sont utilisé pour

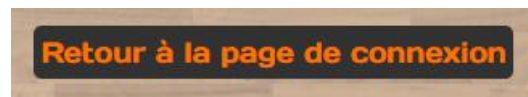
définir "where" et "link". Ces deux propriétés sont attendues par le composant retour. Cela garantit que lorsque l'on utilise ce composant, ces deux propriétés avec les types de données attendues doivent être fournis.

Sur les pages que les utilisateurs peuvent consulter lorsqu'ils ne sont pas connectés, j'ai ajouté le bouton de retour. J'ai utilisé "isLoggedIn" pour vérifier que l'utilisateur n'est pas connecté, et défini "where" et ajoutons le texte à afficher sur le bouton.

Voici le code pour le bouton sur les pages accessibles aux utilisateurs non connectés:

```
<div className='return-btn'>
  {!isLoggedIn && <Retour where="à la page de connexion" link="/login" />}
</div>
```

Voici le bouton de retour:



Barre de navigation + menu déroulant

Pour la barre de navigation, j'ai utilisé "NavLink" un composant fourni par "React-Router". Il permet de créer des liens de navigation avec la capacité de spécifier des styles pour les liens actifs. Grâce à cette fonctionnalité, nous pouvons facilement cibler les liens actifs dans le SCSS et, avec le nom de classe "isActive", changer la couleur de n'importe quel lien actif. Comme on le voit dans le code ci-dessous:

Voici le code qui gère cela:

```
className={({ isActive }) => isActive ? 'menu menu--active' : 'menu'}
.menu {
  padding: 0.5em;
  color: v.$titles;
  &--active{
    color: v.$orange;
  }
}
```



```
const toggleMenu = () => {
  setShowMenu(!showMenu);
};

const closeMenu = () => {
  if (window.innerWidth > 1050) {
    setShowMenu(false);
  }
};

useEffect(() => {
  window.addEventListener('resize', closeMenu);
  return () => {
    window.removeEventListener('resize', closeMenu);
  };
}, []);

const handleLinkClick = () => {
  if (window.innerWidth <= 1050) {
    setShowMenu(false);
  }
};
```

Je utilise cette code pour l'affichage de la menu déroulant. Avec une largeur de plus de 1050px , le menu déroulant ne s'affiche plus et la barre de navigation apparaît.

J'ai réalisé cela en utilisant 'toggleMenu'. Cette fonction "toggles" le menu entre ouvert et fermé. 'closeMenu', ferme le menu lorsque la largeur de l'écran est plus que 1050px. 'useEffect' est utilisé pour écouter les changements de taille de la fenêtre, et vérifié si le menu doit être fermé en fonction de la largeur. 'handleClick' est déclenché lorsqu'un élément de

menu est cliqué, il prends également en compte, la largeur de la fenêtre pour fermer le menu si nécessaire.

```
<nav>
  <div className="hamburger" onClick={toggleMenu}>
    <div className={`icone-burger ${showMenu ? 'close' : 'open'}`}></div>
  </div>

  <div className={`navbar ${showMenu ? 'show-menu' : ''}`>
```

Le menu déroulant et la barre de navigation



J'ai envisagé d'utiliser un package pour le menu déroulant, mais vu que c'est un processus relativement simple, j'ai décidé de le réaliser avec du code brut. Même s'il est peu probable qu'un package de menu déroulant ralentisse le site, il est toujours sage de minimiser les nombre des packages et faire en code brut pour des fonctionnalités simples. Cela réduit la complexité du code et permet une meilleure performance et une maintenance plus facile du site.

Toastify

Pour la gestion des messages d'erreur et des notifications de succès sur notre site, j'ai opté pour l'utilisation de "Toastify", une bibliothèque pour React. J'ai choisi "Toastify", parce que ça nous permet de créer et de configurer des notifications de manière hautement adaptable et dynamique, de manière simple et rapide.

Voici le code pour réaliser les messages:

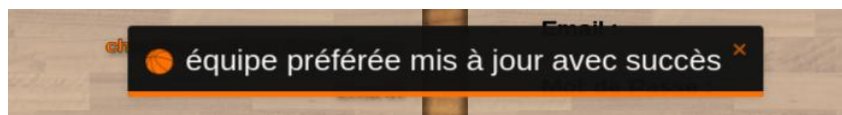
```
import clipartLogo from '../../../assets/design/clipart1224.png';
import 'react-toastify/dist/ReactToastify.css';
import './toastsuccess.scss'

export const toastSuccess = {
  position: 'top-center',
  theme: "dark",
  closeOnClick: true,
  pauseOnHover: true,
  autoClose: 2000,
  progressBar: true,
  icon: <img src={clipartLogo} alt="Success Icon" width="24" height="24" />,
};

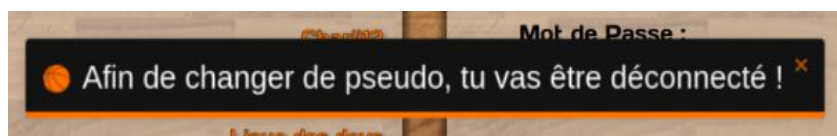
export const toastWarning = {
  position: 'top-center',
  theme: "dark",
  closeOnClick: true,
  pauseOnHover: true,
  autoClose: 2000,
  progressBar: true,
  icon: <img src={clipartLogo} alt="Success Icon" width="24" height="24" />,
};

if (updateComplete) {
  toast.success(`${paramName} mis à jour avec succès`, toastSuccess)
}
```

Voici le toast succès:



Voici le toast warning:



Certaines petites fonctionnalités de style doivent être modifiées. J'aimerais que la barre de progression sur le toast warning soit rouge plutôt qu'orange et que le ballon de basket soit également rouge. Ce sont des changements que je prévois d'apporter dans un avenir proche.

Infos NBA

Cette page était en quelque sorte un casse-tête. J'avais énormément d'informations à mettre sur une seule page, avec un espace limité. L'ensemble du calendrier et de la composition des équipes pour l'ouest et l'est, soit 60 cases au total.

Je suis arrivé à la conclusion que la seule façon d'y parvenir serait de créer une boîte pop-up. J'ai réalisé cela avec le code ci-dessous:

```
const [isRosterPopUpOpen, setRosterPopUpOpen] = useState(false);
const [isSchedulesPopUpOpen, setSchedulesPopUpOpen] = useState(false);
const [isRosterWestPopUpOpen, setRosterWestPopUpOpen] = useState(false);
const [isSchedulesWestPopUpOpen, setSchedulesWestPopUpOpen] = useState(false);

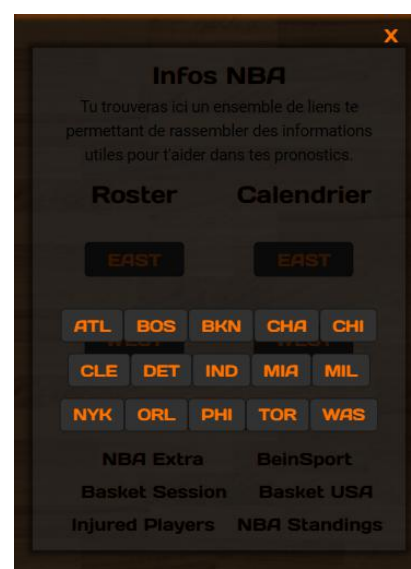
const toggleRosterPopUp = () => {
  setRosterPopUpOpen(!isRosterPopUpOpen);
  setSchedulesPopUpOpen(false);
  setRosterWestPopUpOpen(false);
  setSchedulesWestPopUpOpen(false);
};

<div className="roster-east">
  <button className='pop-up-click' onClick={toggleRosterPopUp}>EAST</button>
  {isRosterPopUpOpen && (
    <div className='pop-up-container'>
      <button className='close-button' onClick={toggleRosterPopUp}>X</button>
      <ul>
        <div className='left-column-re'>
          <li><a onClick={() => openLinkInNewWindow('https://www.nba.com/hawks/roster')} rel="noopener noreferrer">ATL</a></li>
```

Le code total peut être trouvé dans l'annexe "Infos NBA" en page 40 à 41. En utilisant "useState" j'ai attribué à chaque pop-up son propre state pour suivre s'il est était ouvert ou fermé. Lorsque le utilisateur clique sur le bouton, le fonction "togglePop" est déclenché et le pop-up est ouvert.

Chaque lien est ouvert dans une nouvelle fenêtre avec rel="noopener" qui est une mesure de sécurité supplémentaire pour empêcher une fenêtre ouverte avec window.open() d'accéder à window.opener sur la nouvelle fenêtre. Cela réduit le risque d'attaques potentielles telles que les attaques pop-up malveillantes ou le détournement de fenêtres.

Voici le résultat final:



Difficultés Rencontrées

Mon principal problème était de demander de l'aide. Étant fille unique d'une petite famille, J'ai toujours été habitué à agir de manière indépendante, donc travailler en groupe et demander de l'aide plutôt que de trouver une solution moi-même m'était quelque peu étranger. Cependant, au fil de ma formation, j'ai progressivement réalisé que demander de l'aide n'était pas un signe de faiblesse, mais une opportunité d'apprentissage. J'ai dû apprendre à mettre de côté ma fierté pour avancer de manière plus efficace.

Les principales difficultés que j'ai rencontrées provenaient de la tentative de faire fonctionner le mise à jour du profil. J'ai eu du mal pendant un certain temps à savoir où placer les informations entre le reducer, actions et le store. J'ai aussi eu du mal à écrire ma première requête axios mais après la première, les autres étaient plus faciles.

J'ai rencontré aussi un problème en travaillant sur le code de changement de pseudo, qui a nécessité quelques dépannages et l'entraide, pour résoudre. Lors du changement du nom d'utilisateur, cela a entraîné une erreur d'authentification due à un jeton JWT invalide. Pour résoudre cela, j'ai dû mettre en place une solution pour forcer une nouvelle connexion lorsque le nom d'utilisateur (pseudo) était modifié. Cela a été résolu en intégrant la logique suivante dans l'extrait de code fourni ci-dessous:

```
if (pseudo !== loggedUser.username) {
  toast.warning('Afin de changer de pseudo, vous allez être déconnecté !', toastWarning);
  setTimeout(() => {
    document.cookie = 'isLogged=;expires=Thu, 01 Jan 1970 00:00:00 UTC';
    document.cookie = 'userInfo=;expires=Thu, 01 Jan 1970 00:00:00 UTC';
    dispatch(updateUsername(pseudo));
    dispatch(resetStore());
    navigate('/');
  }, 2000);
} else {
  const updateComplete = dispatch(updateUserProfile(updatedUserData));
  if (updateComplete) {
    toast.success(`${paramName} mis à jour avec succès`, toastSuccess)
  }
}
```

J'ai également eu du mal à comprendre le MCD. Ce n'était déjà pas mon sujet le plus fort ou préféré, mais la complexité du le nôtre m'a perdu.

Difficultés Rencontrées en groupe

Nous souhaitions utiliser une API proposant des données en temps réel sur la NBA, déjà disponible en ligne. Cependant, nous n'avons pas trouvé de solution gratuite répondant à nos besoins, alors nous avons dû la créer nous-mêmes.

Où j'aurais pu être meilleure

Entraide: J'aurais pu demander de l'aide plus souvent.

Github: Bien que je travaillais chaque jour sur une nouvelle branche et que je fasse un push chaque soir, je n'ai pas effectué suffisamment de commits ni de merges. Lors-qu'est venu le moment de fusionner ma branche avec la branche dev, j'ai rencontré de nombreux conflits.

Présentation du projet: Je n'ai pas pu participer aux présentations du projet. Malheureusement, ma peur de parler devant un groupe de personnes m'a envahi et je me suis effondré le matin même. C'est quelque chose qui ne me dérange pas en anglais, ce qui rend la situation encore plus frustrante. Malheureusement, l'idée de parler à 30 à 40 autres personnes, dans une langue que je ne suis pas tout à fait sûre de parler, était trop. J'espère progressivement renforcer ma confiance dans ce domaine et pouvoir me confronter à la prise de parole en public, en français, à l'avenir.

Jeu d'essai

Le jeu d'essai est un test conçu pour tester la fonctionnalité, les performances ou la sécurité d'un site Web. Ils sont utilisés pour simuler des situations réelles et vérifier que le site réagit de manière appropriée

Pour garantir le bon fonctionnement de certaines fonctionnalités, nous avons réalisé des tests initiaux sur la création de comptes et la connexion. Étant donné que l'accès à 95% du site nécessite un compte utilisateur, il était impératif que ces deux fonctions soient opérationnelles de manière optimale.

Par la suite, nous avons fait des tests complémentaires au fur à mesure pour être sûrs que toutes les fonctionnalités opéraient correctement. Personnellement j'ai réalisé des tests sur la mise à jour des informations utilisateurs.

Données en entrée

Mes textes se concentraient sur la modification de l'adresse e-mail, du nom d'utilisateur, du mot de passe et de l'équipe préférée des utilisateurs. Les données d'entrée jouent un rôle crucial dans la validation et l'efficacité de ces processus de mise à jour.

Mise à Jour de l'Email: Pour évaluer la fonction de mise à jour de l'email, différents scénarios ont été testés. Les données d'entrée pour ces tests incluaient des adresses email valides et invalides, des adresses déjà utilisées par d'autres utilisateurs, ainsi que des formats d'email variés pour garantir la conformité aux normes.

Mise à Jour du nom d'Utilisateur: Les tests de modification du nom d'utilisateur ont été effectués avec des noms uniques et existants dans la base de données. Il s'agissait de garantir l'unicité du nom d'utilisateur et de valider la fonctionnalité d'édition.

Mise à Jour du changement du Mot de Passe: Les données d'entrée pour les tests de modification du mot de passe comprenaient des mots de passe forts, des mots de passe faibles, des combinaisons de caractères spéciaux, de lettres majuscules et minuscules. Ces tests ont été effectués pour garantir la robustesse du système de gestion des mots de passe.

Mise à Jour de l'Équipe Préférée: Pour la mise à jour de l'équipe préférée, Des tests ont été effectués pour garantir que les données étaient correctement envoyées au Redux Store.

Données attendues

Les tests de modification des informations utilisateur ont été effectués avec des données spécifiques pour chaque cas de mise à jour. Voici les types de données attendues pour chaque processus de modification :

L'email: Les données attendues pour la fonction de mise à jour des e-mails étaient de n'accepter que les adresses e-mail valides au format 'nom@domaine.extension'. Les adresses e-mail invalides ainsi que celles déjà utilisées ne devaient pas être acceptées.

Nom d'Utilisateur: Les données attendues pour la fonction de mise à jour du nom d'utilisateur étaient de n'accepter que noms d'utilisateur uniques, qui ne sont pas déjà stockés dans la base de données.

Mot de Passe: Les données attendues pour la fonction de mise à jour du mot de passe étaient la validation de l'évaluateur de force du mot de passe, fournissant une évaluation de faible, moyen, bon ou fort selon les caractères utilisés.

l'Équipe Préférée: Les données attendues pour la fonction de mise à jour de l'équipe préférée étaient que la sélection d'une équipe valide parmi celles disponibles dans le menu déroulant envoyait les données correctes au Redux Store.

Données obtenues

Tous les résultats attendus étaient les résultats obtenus. Lors de la mise à jour de l'e-mail, du nom d'utilisateur et de l'équipe favorite, les informations étaient affichées sur l'écran à côté et toutes les données étaient correctement mises à jour dans la base de données. Cela était également vrai pour la fonction de mise à jour du mot de passe qui n'était pas affichée à l'écran.

Veille technologique sur la sécurité

Les cyberattaques représentent une menace sérieuse dans le monde d'Internet et exigent une approche sérieuse et prudente. Les données sensibles, stockées sur un site web, sont des cibles pour les personnes malveillantes qui souhaitent les voler et les utiliser de manière préjudiciable. Il est alarmant de constater que près des deux tiers des individus utilisent le même mot de passe pour plusieurs comptes en ligne. Ainsi, un mot de passe utilisé sur votre site web pourrait potentiellement compromettre non seulement la sécurité de votre site, mais également celle de leurs e-mails, comptes bancaires et plus encore.

La surveillance de la sécurité technologique est un élément essentiel de la protection des données contre ces menaces. Elle est en constante évolution et implique donc une surveillance constante des nouvelles cyberattaques, des nouvelles vulnérabilités de sécurité et des solutions innovantes pour contrer ces risques. Cela nécessite d'être à jour sur les pratiques de cryptage, les protocoles de sécurité et les meilleures stratégies de protection des données.

Sécurisation de DMFC

Pour renforcer la sécurité de notre site et assurer la protection de nos données, nous avons déployé plusieurs méthodes de sécurité. Une partie essentielle de notre approche a été la mise en place d'une logique de sécurité qui limite l'accès aux différentes fonctionnalités en fonction du rôle de l'utilisateur connecté. Cette sécurisation a été réalisée en combinant plusieurs outils et techniques, notamment l'utilisation de React Router pour la gestion des routes, Axios pour les appels API sécurisés, la gestion des cookies pour la préservation des sessions, et la vérification des rôles attribués à chaque utilisateur. De plus, pour sécuriser notre API, nous avons mis en place l'utilisation d'un jeton JWT (JSON Web Token) pour l'authentification des requêtes. Cette méthode d'authentification basée sur des jetons offre un niveau supplémentaire de sécurité en attribuant des jetons uniques et cryptés à chaque utilisateur autorisé, renforçant ainsi la sécurité lors des échanges de données entre le frontend et l'API. J'ai aussi utilisé "rel=noopener norefferrer" sur tout les liens pour réduire les risques de sécurité liés à l'ouverture de nouvelles fenêtres.

Sites anglophone

Utilisation de sites anglophones

Étant donné que l'anglais est ma langue maternelle, l'utilisation des sites web en anglais ne pose aucun problème pour moi. Cela a été particulièrement utile lors de la recherche de packages React, car j'ai vu que très peu de documentation est disponible en français. Bien que le plugin Google Translate pour le navigateur Chrome soit extrêmement utile, ses capacités de traduction sont loin d'être parfaites. Il est vrai que la plupart des sites web liés au développement web sont principalement en anglais, et les ressources disponibles dans cette langue sont souvent complexes et détaillées. Cependant, cela ne constitue pas un obstacle pour moi, car je suis capable de naviguer, de lire et de comprendre les informations techniques dans cette langue. Je me considère très chanceuse d'avoir l'anglais comme langue maternelle, ce qui me permet d'approfondir mes connaissances dans le domaine du développement web.

Recherche sur un site anglais

En examinant les sécurités et les vulnérabilités de ReactJS, je suis tombé sur ce site, www.logrocket.com, pour des analyses frontales. Il explique en détail les avantages de l'utilisation de l'authentification de connexion côté serveur. Une méthode que nous mettons en pratique sur notre site internet pour gérer la sécurité.

Extrait d'un site anglophone

There are several uses and benefits to using server-side login authentication. First and foremost, it helps ensure the security of the server and the resources it contains. By requiring users to provide a username and password, the server can verify that the person attempting to log in is authorized to do so. Another benefit of server-side login authentication is that it can provide a centralized system for managing user accounts. This means that the server administrator can easily add, remove, or update user accounts, and the changes will be immediately reflected across the entire server. This can make it easier to manage user access and ensure that only authorized users have access to the server and its resources. In addition, server-side login authentication can provide a way for users to securely access resources from any device — as long as they have the necessary credentials. This can be especially useful for organizations with multiple locations or employees needing to access resources remotely.

Overall, server-side login authentication is a useful and important tool for protecting the security of servers and the resources they contain. It can help prevent unauthorized access and provide a centralized system for managing user accounts, making it an essential component of any secure server environment.

Traduction

Il y a plusieurs usages et avantages de l'utilisation de l'authentification de connexion côté serveur. Tout d'abord, ça aide à assurer la sécurité de le serveur et les ressources qu'il contient. En obligeant les utilisateurs de fournir un nom d'utilisateur et un mot de passe, le serveur peut vérifier que la personne qui essaye de se connecter, est autorisée de le faire. Un autre avantage de l'authentification de connexion côté serveur est qu'il peut fournir un système centralisé pour la gestion des comptes utilisateurs. Cela signifie que l'administrateur peut facilement ajouter, supprimer ou mettre à jour les comptes d'utilisateur et les modifications seraient reflétées immédiatement sur le serveur entier. Cela peut faciliter la gestion de l'accès des utilisateurs et garantir que seuls les utilisateurs autorisés ont accès de le serveur et ses ressources. En plus, l'authentification de connexion côté serveur peut fournir un moyen pour les utilisateurs d'accéder, en toute sécurité, des ressources depuis n'importe quel appareil, tant qu'ils ont des identifiants nécessaires. Cela peut-être particulièrement utile pour les entreprises avec plusieurs lieux différents ou des employés qui ont besoin d'accéder aux ressources à distance. Globalement, l'authentification de connexion côté serveur est un outil très utile et importante pour la protection et la sécurité des serveurs et des ressources qu'ils contiennent. Il peut aider à empêcher l'accès non autorisé et fournir un système centralisé de gestion des comptes des utilisateurs, cela en fait un composant essentiel de tout environnement de serveur sécurisé.

Conclusion

Même si j'avais plus que peur de commencer ce projet d'un mois, ce fut une courbe d'apprentissage incroyable. C'était une opportunité de plonger tête baissée dans un travail indépendant, ce qui nous a énormément appris. Tout au long de la formation on n'a jamais vraiment eu de feuille blanche. Il y avait toujours un modèle existant ou un projet déjà commencé sur lequel travailler. Cependant, repartir de zéro, sans aucun point de départ, a été initialement intimidant mais incroyablement gratifiant. Chaque semaine qui passait, je me sentais de plus en plus confiante. J'ai pu aborder le développement et résoudre les problèmes rencontrés en cours de route avec certitude, avec un sentiment de capacité et des connaissances de base toujours croissantes.

La mise en pratique de nos apprentissages a été une expérience exceptionnelle. Nous avons été surpris par le fait que nous avions besoin de peu d'aide de la part des formateurs. À chaque fois qu'un défi se présentait, il semblait qu'un membre de notre équipe avait une solution à proposer. Cette expérience a montré les immenses avantages du travail en équipe, où chacun peut contribuer avec ses connaissances pour résoudre les problèmes. Cela m'a appris non seulement à travailler en groupe, mais aussi à apprécier et en profiter.

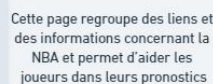
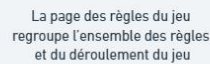
J'ai vraiment apprécié non seulement le côté codage, mais aussi la recherche de technologies et de bibliothèques. Ce processus a été très enrichissant et les journées passaient rapidement. Le débogage a également été très gratifiant. Trouver des solutions à des problèmes est ce que j'apprécie le plus et c'est ce qui m'a motivé à me lancer dans le domaine de développement web.

L'avenir

Mes projets pour l'avenir sont de passer mon titre professionnel et continuer d'apprendre et de développer mes connaissances dans le domaine du développement web. J'ai un abonnement d'un an à www.oreilly.com qui est une plateforme d'apprentissage en ligne, où je prends des cours quand je peux. J'envisage également d'apprendre Symfony, grâce aux replays et supports de cours fournis par O'clock. Après deux mois de très peu de développement backend, je dois le reprendre pour être à nouveau à un niveau compétent. Je code pendant mon temps libre, pour développer mon portfolio personnel et je suis à la recherche d'un stage ou d'une alternance et j'espère ensuite trouver un poste à temps plein dans une entreprise où je pourrai bénéficier de l'expérience de travailler aux côtés de développeurs web expérimentés.

Wireframes





General stats

LOGO Profil Stat Général Ranking Général Stats NBA Cheat Sheet

Joueur 1 Stats

Ligue : Ligue des Justiciers
Titre : MVP

Saison en cours :

Pronostique réalisé : 16/24
Nombre de points : 27 points
Classement Global : **1ère position**

Stats All Time :

Saison joué : 3
Nombre de Pronostique : 24
Pronostique correcte : 2
Nombre de point dernière saison : 24 points
Nombre de point maximale en 1 saison : 68 points

Début de saison : XX/XX/XXXX
Fin de Saison : XX/XX/XXXX

Infos globales saison

Si osculatur puer tuus aut uxorem tuam, osculam, non dico quod omnia que sunt hominis, et sic non tangetur, si aut eis moriatur.

Quando ambulabat agendis admonere te qualis actio. Si ad corpus, que plerumque imaginare tecum in balneo quidam aquam fundes aliquid discrimen vituperis usum alii furantur.

Nom de Joueur	Nombre de pts
Joueur 1 ++2	27 Points
Joueur 2 ++1	24 Points
Joueur 3 +-2	22 Points
Joueur 4 =	18 Points

DMC Rules Conditions générales

La page des statistiques regroupe l'ensemble des statistiques du (ou un) joueur

General Stats

LOGO

Joueur 1 Stats

Ligue : Ligue des Justiciers
Titre : MVP

Saison en cours :

Pronostique réalisé : 16/24
Nombre de points : 27 points
Classement Global : **1ère position**

Stats All Time :

Saison joué : 3
Nombre de Pronostique : 24
Pronostique correcte : 2
Nombre de point dernière saison : 24 points
Nombre de point maximale en 1 saison : 68 points
Nombre de saison gagné : 2

DMC Rules Conditions générales

General Ranking

LOGO Profil Stat Général Ranking Général Stats NBA Cheat Sheet

Nom de Joueur	Nombre de pts
Joueur 1 ++2	27 Points
Joueur 2 ++1	24 Points
Joueur 3 +-2	22 Points
Joueur 4 =	18 Points
Joueur 5 =	17 Points
Joueur 6 =	14 Points
Joueur 7 =	8 Points
Joueur 8 =	6 Points
Joueur 9 =	0 Points
Joueur 10 =	0 Points

Début de saison : XX/XX/XXXX
Fin de Saison : XX/XX/XXXX

Infos globales saison

Si osculatur puer tuus aut uxorem tuam, osculam, non dico quod omnia que sunt hominis, et sic non tangetur, si

DMC Rules Conditions générales

Au clic sur le logo → retour à la page news

Classement général regroupant tous les joueurs de la ligue. Au clic sur une ligne → accès aux stats du joueur

General Ranking

LOGO

La Ligue des Justiciers

Nom de Joueur	Nombre de pts
Joueur 1 ++2	27 Points
Joueur 2 ++1	24 Points
Joueur 3 +-2	22 Points
Joueur 4 =	18 Points
Joueur 5 =	17 Points
Joueur 6 =	14 Points
Joueur 7 =	8 Points
Joueur 8 =	6 Points
Joueur 9 =	0 Points
Joueur 10 =	0 Points

DMC Rules Conditions générales

Header

```
import { Link } from 'react-router-dom';
import { useSelector } from 'react-redux';

import dmfcLogo from '../../../assets/design/dmfc-logo.png';
import clipartLogo from '../../../assets/design/clipart1224.png';

import './Header.scss';

function Header() {
  const isLoggedIn = useSelector((state) => state.user.isLoggedIn);

  return (
    <header className='header'>
      <div className="logo-one">
        <img src={dmfcLogo} alt="logo-1" />
      </div>
      <h1 className='header-title'>{isLoggedIn ? <Link to="/">DMFC - Game</Link> : "DMFC - Game"}</h1>
      <div className="logo-two">
        <img src={clipartLogo} alt="logo-2" />
      </div>
    </header>
  )
}
```

export default Header

```
@use '../../../styles/variables' as v;

.header {
  position: fixed;
  z-index: 2;
  top: 0;
  width: 100%;
  background: url('/src/assets/design/wood-black.png');
  text-align: center;
  color: v.$titles;
  display: flex;
  align-items: center;
  height: 2em;
  justify-content: left;
  padding: 0 20px;
  font-family: v.$main-font;
  font-size: 3em;
  box-shadow: 0 5px 4px 0 v.$light-orange;
  h1 {
    margin: 0 20px;
  }
  img {
    height: 1.8em;
    margin-top: 0.1em;
  }
  .logo-two{
    display: none;
  }

  @media (min-width: 768px) {
    justify-content: space-evenly;
    .logo-two{
      display: unset;
    }
    .logo-one, .logo-two {
      width: 116px;
    }
  }
}
```


Mettre à jour le profil

```
const handleInput = (event) => {
  dispatch(setInputValue(event.target.id, event.target.value));
};

const handleSubmit = async (event, paramName) => {
  event.preventDefault();
  let isErrorMail = false;
  if (email !== loggedUser.email) {
    isErrorMail = verifyMail(email);
    dispatch(setMailError(isErrorMail));
  }
  if (!isErrorMail) {
    const updatedUserData = {
      username: pseudo,
      email: email,
      password: password,
      team: team,
    };
    if (pseudo !== loggedUser.username) {
      toast.warning('Afin de changer de pseudo, vous allez être déconnecté !', toastWarning);
      setTimeout(() => {
        document.cookie = `isLogged=;expires=Thu, 01 Jan 1970 00:00:00 UTC`;
        document.cookie = `userInfos=;expires=Thu, 01 Jan 1970 00:00:00 UTC`;
        dispatch(updateUsername(pseudo));
        dispatch(resetStore());
        navigate('/');
      }, 2005);
    } else {
      const updateComplete = dispatch(updateUserProfile(updatedUserData));
      if (updateComplete) {
        toast.success(`${paramName} mis à jour avec succès`, toastSuccess)
      }
    }
  }
};

const handleLeagueChange = event => {
  event.preventDefault();
  dispatch(updateUserProfile({ league: league }));
}
```

```
...
<form className="change-info" onSubmit={handleSubmit}>
  <h3>Changer mes Infos</h3>
  <div className="changeContainer">
    <input label="Email :" htmlFor="email" id="email" type="email" className="inputContainer" value={email} onChange={handleInput} placeholder="changer ton email" required={true}/>
    {mailError && <p className="error-message">Le format du mail n'est pas correct.</p>}
    <div className="form-btn">
      <button type="submit" onClick={(event) => handleSubmit(event, 'email')}>Changer</button>
    </div>
  </div>
  <div className="changeContainer" id="passwordInput">
    <input className="password inputContainer" label="Mot de Passe :" htmlFor="mot de passe" id="password" type="password" value={password} onChange={handleInput} placeholder="mot de passe"/>
    <div className="form-btn">
      <button type="submit" onClick={(event) => handleSubmit(event, 'mot de passe')}>Changer</button>
    </div>
  </div>
  <div className="changeContainer">
    <input label="Pseudo :" htmlFor="pseudo" id="pseudo" type="text" className="inputContainer" value={pseudo} onChange={handleInput} placeholder="change ton pseudo"/>
    <div className="form-btn">
      <button type="submit" onClick={(event) => handleSubmit(event, 'pseudo')}>Changer</button>
    </div>
  </div>
  <div className="changeContainer">
    <div className="inputContainer">
      <label>Équipe Préférée : </label>
      <select name="team" id="team" onChange={handleInput} value={team}>
        <option value="">Changer ta équipe préférée</option>
        {teamOptions}
      </select>
    </div>
    <div className="form-btn">
      <button type="submit" onClick={(event) => handleSubmit(event, 'équipe préférée')}>Changer</button>
    </div>
  </div>
</form>
...
```

Infos NBA

```
const [isRosterPopUpOpen, setRosterPopUpOpen] = useState(false);
const [isSchedulesPopUpOpen, setSchedulesPopUpOpen] = useState(false);
const [isRosterWestPopUpOpen, setRosterWestPopUpOpen] = useState(false);
const [isSchedulesWestPopUpOpen, setSchedulesWestPopUpOpen] = useState(false)
```

```
const toggleRosterPopUp = () => {
  setRosterPopUpOpen(!isRosterPopUpOpen);
  setSchedulesPopUpOpen(false);
  setRosterWestPopUpOpen(false);
  setSchedulesWestPopUpOpen(false);
};
```

```
const toggleSchedulesPopUp = () => {
  setSchedulesPopUpOpen(!isSchedulesPopUpOpen);
  setRosterPopUpOpen(false);
  setRosterWestPopUpOpen(false);
  setSchedulesWestPopUpOpen(false);
};
```

```
const toggleRosterWestPopUp = () => {
  setRosterWestPopUpOpen(!isRosterWestPopUpOpen);
  setRosterPopUpOpen(false);
  setSchedulesPopUpOpen(false);
  setSchedulesWestPopUpOpen(false);
};
```

```
const toggleSchedulesWestPopUp = () => {
  setSchedulesWestPopUpOpen(!isSchedulesWestPopUpOpen);
  setRosterPopUpOpen(false);
  setSchedulesPopUpOpen(false);
  setRosterWestPopUpOpen(false);
};
```

```
const openLinkInNewWindow = (url) => {
  window.open(url, '_blank');
};
```

```
<div className='pop-up-boxes'>
  <div className="rosters">
    <div className="roster-east">
      <button className='pop-up-click' onClick={toggleRosterPopUp}>EAST</button>
      {isRosterPopUpOpen && (
        <div className='pop-up-container'>
          <button className='close-button' onClick={toggleRosterPopUp}>X</button>
          <ul>
            <div className='left-column-re'>
              <li><a onClick={() => openLinkInNewWindow('https://www.nba.com/hawks/roster')} rel="noopener noreferrer">ATL</a></li>
              <li><a onClick={() => openLinkInNewWindow('https://www.nba.com/celtics/roster')} rel="noopener noreferrer">BOS</a></li>
              <li><a onClick={() => openLinkInNewWindow('https://www.nba.com/nets/roster')} rel="noopener noreferrer">BKN</a></li>
              <li><a onClick={() => openLinkInNewWindow('https://www.nba.com/hornets/roster')} rel="noopener noreferrer">CHA</a></li>
              <li><a onClick={() => openLinkInNewWindow('https://www.nba.com/bulls/roster')} rel="noopener noreferrer">CHI</a></li>
            </div>
          </ul>
          <div className='middle-column-re'>
            <li><a onClick={() => openLinkInNewWindow('https://www.nba.com/cavaliers/roster')} rel="noopener noreferrer">CLE</a></li>
            <li><a onClick={() => openLinkInNewWindow('https://www.nba.com/pistons/roster')} rel="noopener noreferrer">DET</a></li>
            <li><a onClick={() => openLinkInNewWindow('https://www.nba.com/pacers/roster')} rel="noopener noreferrer">IND</a></li>
            <li><a onClick={() => openLinkInNewWindow('https://www.nba.com/sixers/roster')} rel="noopener noreferrer">PHI</a></li>
            <li><a onClick={() => openLinkInNewWindow('https://www.nba.com/bucks/roster')} rel="noopener noreferrer">MIL</a></li>
          </div>
        </ul>
        <div className='right-column-re'>
          <li><a onClick={() => openLinkInNewWindow('https://www.nba.com/knicks/roster?icmp=int_knicks_roster_parent_nav_190514')} rel="noopener noreferrer">NYK</a></li>
          <li><a onClick={() => openLinkInNewWindow('https://www.nba.com/magic/roster')} rel="noopener noreferrer">ORL</a></li>
          <li><a onClick={() => openLinkInNewWindow('https://www.nba.com/sixers/roster')} rel="noopener noreferrer">PHI</a></li>
          <li><a onClick={() => openLinkInNewWindow('https://www.nba.com/raptors/roster')} rel="noopener noreferrer">TOR</a></li>
          <li><a onClick={() => openLinkInNewWindow('https://www.nba.com/wizards/roster')} rel="noopener noreferrer">WAS</a></li>
        </div>
      </ul>
    </div>
  )}
</div>
```



```

.paragraph {
  line-height: 1.5em;
  text-align: center;
  margin-top: 0.5em;
  margin-bottom: 1em;
}

.titles {
  display: flex;
  justify-content: space-evenly;
}

.usefull-links {
  margin: 1em;
}

.links {
  display: flex;
  justify-content: space-evenly;
  flex-wrap: wrap;
}

.pop-up-boxes {
  display: flex;
  justify-content: space-evenly;
}

.pop-up-click {
  border: 1px solid v.$shadow;
  color: v.$orange;
  background-color: v.$nav-back;
  padding: 0.6em;
}

.roster-east, .roster-west, .schedules-east, .schedules-west {
  display: block;
  margin: 1.5em;

  .left-column-re, .left-column-rw, .left-column-se, .left-column-sw {
    display: flex;
    align-self: flex-start;
    margin: 0.2em;
  }

  .middle-column-re, .middle-column-rw, .middle-column-se, .middle-column-sw {
    display: flex;
    align-self: center;
    margin: 0.2em;
  }
}

```

```

.right-column-re, .right-column-rw, .right-column-se, .right-column-sw {
  display: flex;
  align-self: flex-end;
  margin: 0.2em;
}

.left-column-re, .left-column-rw, .left-column-se, .left-column-sw,
.middle-column-re, .middle-column-rw, .middle-column-se, .middle-column-sw,
.right-column-re, .right-column-rw, .right-column-se, .right-column-sw {
  li {
    a {
      display: block;
      padding: 0.5em 0.75em;
      border: 1px solid v.$shadow;
      margin-bottom: 0.3em;
      text-align: center;
      text-decoration: none;
      color: v.$orange;
      background-color: v.$nav-back;
      border-radius: 0.3em;
      cursor: pointer;
      font-family: v.$main-font;
    }
  }
}

```

```

.pop-up-container {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background: v.$shadow-dark;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  z-index: 1;
  caret-color: v.$orange;
}

.close-button {
  position: sticky;
  background: transparent;
  border: none;
  font-size: 1.2em;
  cursor: pointer;
  color: v.$orange;
  right: 0.5em;
  top: 19em;
  font-family: "Days One";
}

```

```

@media screen and (min-width : 768px) {
  .pop-up-click {
    padding-right: 1.2em;
  }
}

```