

## Table des matières

Introduction	2
Lancement du projet	3
Cahier des charges	4
Je m'invente des vies	4
Il faut commencer sur de bonnes bases	6
Le design	8
Le choix d'une palette	8
Il y a du boulot pour le logo	9
La première fois c'est toujours difficile	10
Il y a encore du travail	15
On en voit le bout	18
Intégration	23
Lara vèle	23
Un sac à dos bien rempli	23
De l'autre côté du sac à dos	37
Des navigateurs sans Javascript, ça existe encore ?	42
Il est venu le temps du Javascript	46
Ajax, une formule sans recharge	59
Conclusion	65
Bibliographie	66
Remerciements	67

## Introduction

Dans le but de finaliser ma formation en techniques graphiques orientation techniques infographiques, il m'a été demandé de réaliser un projet de fin d'études. Le but de ce projet est de démontrer le niveau d'acquisition de mes compétences à mes professeurs mais aussi à un jury de professionnels extérieurs à ma formation.

Je suis membre actif du Royal Basket Club de Ciney depuis maintenant 20 ans. Je me suis naturellement lancé pour défi de réaliser un projet qui me tient à cœur : la refonte du site internet de mon club. Mon but personnel est de repousser mes limites, d'étoffer mes compétences et d'améliorer mes connaissances. Ce projet est d'une plus grande ampleur que ce que j'avais fait jusqu'à présent : réaliser en Laravel, une technologie qui m'était encore inconnue à l'époque.

Le site existant était obsolète, le design avait besoin d'être rafraîchi et je ne pouvais pas me baser sur l'administration déjà mise en place sachant qu'elle était protégée par son propriétaire. J'avais donc tout à refaire.

## Lancement du projet

La première étape consiste à rencontrer le client, Jean-Paul Houart, secrétaire du Club. Jean-Paul est la personne qui gère le site depuis toujours mais ce n'est pas lui qui l'a créé. Il est toujours partant quand il s'agit de se montrer à la pointe de la technologie ou d'améliorer le fonctionnement de son club, surtout quand la main d'oeuvre est offerte.

Lorsque je lui ai présenté mon projet, il était bien entendu enchanté et avait déjà en tête beaucoup de fonctionnalités qu'il voulait implémenter sur le nouveau site. J'ai donc organisé une réunion dans le but de savoir exactement ce qu'il attendait du nouveau site du club. Il m'a uniquement parlé de l'administration du site, ce qu'il utilise exclusivement. J'avais donc le champ libre pour ce qui était du site utilisé par les visiteurs, ce qui m'enchantait.

Pour ce qui était de l'administration, il avait cependant beaucoup de contraintes. Il voulait une administration des utilisateurs orientée sur les familles plutôt que les utilisateurs individuels, il voulait pouvoir gérer facilement tous les matchs du club, et surtout pouvoir en ajouter à partir d'un fichier Excel sachant qu'en début de saison il doit encoder près de 300 matchs pour la totalité du club. Il souhaitait ajouter un système de pub pour les sponsors ainsi qu'une gestion des événements organisés par le club pour en faire la promotion sur le site, tout cela bien sûr, en conservant ce que l'ancien site proposait déjà.

Jean-Paul ne voulait pas s'arrêter là! Il était très intéressé par mon projet de fin d'études mais ce qui l'intéressait encore plus c'était une promesse de collaboration à long terme. Il avait beaucoup d'idées à mettre en place, susceptibles d'amener d'autres fonctionnalités sur le site à l'avenir. C'est pourquoi il était plus intéressé par le long terme que par une intervention spontanée de ma part. En aucun cas cela ne me posait de problème, sachant que j'ai toujours eu l'intention de m'investir personnellement dans la vie du club.

Notre premier entretien s'est très bien déroulé, mon but était atteint. J'avais une idée plus précise de ce qu'il voulait et je pouvais débuter mon projet.

## Cahier des charges

### Je m'invente des vies

Pour commencer, j'ai rédigé la liste des raisons ou des besoins pour lesquels une personne vient visiter le site du club. Pour cela, j'ai dû me mettre dans la peau de personnes fictives ayant un objectif et un contexte. J'ai donc créé 7 profils différents:



**Cersei Lannister - 43 ans**

Maman de 3 joueurs au RBC Ciney



**Petyr Baelish - 38 ans**

Membre du conseil d'administration du RBC Ciney



**John Snow - 22 ans**

Coach de l'équipe P1 Hommes au RBC Ciney



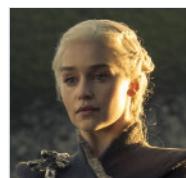
**Jorah Mormont - 43 ans**

Supporter de l'équipe R2 Dames où joue l'amour de sa vie



**Iron Bank - 34 ans**

Sponsor du RBC Ciney depuis 2 ans



**Daenerys Targaryen - 22 ans**

Joueuse dans l'équipe de R2 Dames au RBC Ciney



**Stannis Baratheon - 45 ans**

Papa d'une petite fille qui brûle d'envie de faire du basketball

Une fois ces profils créés, je me suis mis dans la peau de chacun d'entre eux et j'ai établi une série de situations qui leur étaient propres. Le but de cette manœuvre est de définir des besoins chez les futurs utilisateurs de mon site pour ensuite mettre en place les fonctionnalités qui pourront répondre à leurs besoins.

Cet exercice n'est pas facile à optimiser mais il a son importance : s'il est bien fait, cela me permet de cibler avec précision les fonctionnalités à mettre en place et déterminer celles qu'il faut mettre en avant.

### Situations propres aux profils mis en place :

#### Cercei :

- Ma fille étant la plus belle du monde, j'aimerais voir les photos qui ont été prises quand elle joue pour les montrer à ma famille et mes amis.
- Si j'ai un problème j'aimerais trouver facilement la bonne personne du comité à contacter.
- J'aimerais connaître la composition du comité du club et quels sont les membres décisionnaires.
- J'ai perdu la feuille avec les horaires des entraînements et je n'ai pas envie de me déplacer jusqu'à la salle juste pour ça. Je me rends donc sur le site pour trouver l'information.

#### Peter :

- Je veux que les informations de location de la salle soit présentes sur le site.
- Je veux que les fichiers d'assurances, mutualités, déclarations d'accident et autres soient disponibles.

#### John :

- Pas mal de parents de joueurs me demandent où ils peuvent trouver les fichiers d'inscription
- J'aimerais pouvoir partager les fichiers et informations avec les autres coaches.

#### Jorah :

- J'aimerais trouver assez vite tous les matchs qui vont être joués le week-end prochain.
- J'aimerais avoir accès aux résultats des matchs du week-end, et de mon équipe favorite.
- J'aimerais avoir accès facilement à la page de mon équipe favorite.

#### Iron :

- En tant que sponsor du club j'aimerais me retrouver sur le site internet.

#### Daenerys :

- Je me rends sur le site d'un club pour trouver l'adresse de la salle. J'ai souvent le réflexe de chercher l'itinéraire une fois cette adresse trouvée.
- Je me rends sur le site du club pour y trouver le calendrier des matchs de mon équipe ou celui des autres.

#### Stannis :

- Je suis à la recherche d'un club de basket pour ma fille, j'aimerais pouvoir me faire une idée de la mentalité du club et savoir à quoi je m'engage.

## Il faut commencer sur de bonnes bases

Une fois les besoins identifiés, je me suis attaqué à la structure du site et ses fonctionnalités. C'est à cette étape que j'ai décidé quelles pages j'allais intégrer et quelles fonctionnalités j'allais implémenter dans chacune d'elles.

### Structure mise en place :

#### Page d'accueil :

- Slogan et photo du club en bannière
- Calendrier du match du week-end prochain (avec possibilité de filtrer)
- Lien vers les résultats
- News
- Sponsors
- Dernières photos ajoutées

#### Footer :

- Logo du club
- Adresse du club
- E-mail de contact
- Lien vers la page Facebook

#### Page « Équipes » :

- Liste des équipes, chacune renvoyant vers la page dédiée

#### Page « Équipe » :

- Calendrier des matchs à venir
- Résultats des matchs
- Liste des joueurs, entraîneur et assistant de l'équipe
- Horaires d'entraînements
- Photos liées à l'équipe

#### Page « Entraineurs » :

- Liste des fiches des entraîneurs, chacune renvoyant vers la page profil de celui-ci
  - Fiches: photo, nom, prénom, tél, mail, équipe(s) entraînée(s)

#### Page « Comité » :

- Liste des fiches des membres du comité, chacune renvoyant vers la page profil
  - Fiches: photo, nom, prénom, tél, mail, fonction(s)

**Page « Complexe » :**

- Galerie photos du complexe sportif
- Adresse du complexe sportif
- Carte d'accès (Google map)
- Lien vers l'itinéraire
- Tarif de location du complexe sportif et buvette
- Explications des locations

**Page « Contact » :**

- Formulaire de contact
- Informations de contact (adresse et mail)
- Carte d'accès (Google map)
- Lien vers l'itinéraire

**Page « Photos » :**

- Liste des albums photos disponibles, chacun renvoyant vers la page dédiée à celui-ci

**Page « Album » :**

- Galerie des photos de l'album (mur de photos, cliquable pour visualiser en taille réelle)

**Page « Calendrier » :**

- Liste de tous les matchs (avec possibilité de filtrer)
- Liste de tous les résultats (avec possibilité de filtrer)
- Liste des activités du club

**Page « Règlements » :**

- Règlements propres au club avec navigation interne

**Page « Téléchargements » :**

- Fichiers à télécharger (Inscription, Certificat médical, Attestations, ...)

**Page « Coaching » :** (réservée aux coaches)

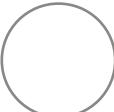
- Liste des fichiers/liens partagés par les coaches du club
- Formulaire pour ajouter un fichier à partager ou un lien (ou les deux)

Une fois la structure du site établie, mes idées étaient plus claires et je pouvais enfin me lancer sur le design visuel.

## Le design

### Le choix d'une palette

La première décision à prendre c'est de choisir une palette de couleur. C'est ce qui va définir tout le reste du design et donner une première identité à mon projet.

	Bleu (principale) #36769E		Orange (secondaire) #F58020
	Bleu foncé #105885		Orange foncé #DC6C11
	Bleu clair #323335		Noir #323335
	Blanc #FFFFFF		Gris #6D6D6D

Sachant que les couleurs de notre club sont le bleu et blanc, j'ai donc choisi le bleu comme couleur principale pour garder l'identité du club à travers le site. Pour ce qui est de la secondaire, je voulais trouver une couleur en accord avec la première et en rapport avec le basketball. Mon choix s'est donc tourné vers l'orange, couleur complémentaire du bleu avec une forte connotation au basketball.

Ces deux choix convenaient très bien pour tout objet graphique mais manquaient un peu de contraste pour une utilisation textuelle. J'ai donc sélectionné deux dérivés plus foncés de ces couleurs avec un contraste suffisant pour la lecture.

Concernant les couleurs de base, j'ai choisi un blanc pur contrairement au noir. Je trouve personnellement qu'un noir pur donne un aspect « cheap » ce que je voulais éviter. J'ai ensuite choisi un gris pour quelques éléments graphiques ainsi qu'un bleu clair pour des éléments plus discrets.

## Il y a du boulot pour le logo

Le logo que le club utilise en ce moment date de quelques années et ne remplit plus les qualités requises pour un logo actuellement. J'ai donc proposé à Jean-Paul de le changer pour qu'il corresponde à certaines normes telles que, transformable en noir et blanc sans perte de qualité, imprimable en toute taille sans perte d'informations, lisible en petite taille. Et personnellement, je souhaitais le moderniser un peu.

Jean-Paul n'était pas contre l'idée de changement malgré que cela impliquerait de changer tout le pôle de communication du club. Il voulait cependant garder le thème ballon & collégiale pour préserver l'attachement du club à la ville de Ciney.

Mon premier réflexe a été d'aller voir les logos des équipes professionnelles de basket. Cela m'a vite donné une idée de ce qui se faisait et vers quoi je devais me diriger. Après avoir sélectionné ceux que je trouvais intéressants, je me suis lancé dans la création de mon logo. J'ai bien entendu utilisé les couleurs que j'avais choisies auparavant et respecté les souhaits de Jean-Paul.

Ancien logo



Nouveau logo



## La première fois c'est toujours difficile

Une fois le logo réalisé, je me suis attaqué au design desktop. J'ai donc repris la structure que j'avais mise en place et je me suis mis à créer un visuel sur sketch.

Sketch n'est pas un programme que nous avons pris en main lors de notre formation mais Mr. Delnatte nous en avait parlé lors de son cours de *design web*. N'étant pas un grand fan de photoshop pour le design web j'ai donc décidé de me lancer sur sketch. Une décision qui a porté ses fruits lors de mon stage chez Exit Studio et pour ce projet. J'ai adopté ce programme et je ne suis pas prêt d'en changer.

J'ai montré le design que j'avais créé à mon professeur Mr. Vilain qui donnait cours de *projet web*. Mon projet de fin d'études faisait partie de ce cours à part entière et je devais en présenter l'avancé ainsi que d'autres lors du jury de la session de janvier. Mr. Vilain m'a donc fait quelques remarques pour améliorer mon design avant de le présenter au jury.

### Design présenté lors du jury de janvier :

*Design de la page d'accueil*

The screenshot shows the main homepage of the Royal Basket Club de Ciney website. At the top, there's a navigation bar with links for Accueil, Club, Calendrier, Multimédia, Complexe, Contact, Coaching, and Connection. Below the navigation is a large banner featuring a team photo. Underneath the banner are links for Conseil d'admin, Albums photos, Entraineurs, Équipes, and Règlements. The main content area includes a "Voir tous nos matchs" section with a table of matches, a "Matchs de la semaine" section, a "Dernière news" section with a "CHRISTMAS BASKET" graphic, and a "Résultats de la semaine" section. There's also a "Nos partenaires" section with logos for Adeps and Ac.

*Suite du design de la page d'accueil*

This screenshot shows a section of the website dedicated to photos. It features a heading "Nos derniers albums photos" with several thumbnail images of basketball teams. Below this is a "Toutes nos photos" link. To the right, there's a sidebar with text about a recent match, a note about a father and son playing together, and social media links for Facebook and Twitter.

*Design de la page de connexion*

This screenshot shows the login page. At the top, it has the same navigation bar as the homepage. Below it is a "Vous connecter" section with fields for "Identifiant" and "Mot de passe". A "Connection" button is present. Below the form, there's a message about incorrect credentials. At the bottom, there are links for "Nous trouver", "Nous contacter", "Nous suivre", and a Facebook icon.

Réalisation du site du Royal Basket Club de Ciney en Laravel

## *Design de la page du calendrier*

## *Design de la page des règlements*



Accueil Club Calendrier Multimédia Complexe Contact Coaching

Connexion

Règlements

## Nos règlements

### Règlement d'ordre intérieur

#### Introduction

Vous trouverez ci-dessous le règlement de discipline sportive du club. Une première lecture vous laissera sans doute une impression de régime sévère, de « dictature »... Par la suite, vous seriez toutefois « obligés » d'admettre que tout ce qui y figure est logique et nécessaire au bon fonctionnement d'un club qui se veut organisé.

Quelques envois de messages nous insistons sur le fait que ce règlement sera de stricte application en toutes circonstances.

#### Entraînements

La participation aux entraînements est **indispensable**.

Le joueur qui pour une raison sérieuse (cas de force majeure, maladie, étude, blessure, travail,...) n'est pas en mesure de participer à un entraînement devra en informer son entraîneur **au plus tard 2 heures** avant le début de celui-ci. Il est vivement conseillé d'arriver **1/4 d'heure avant le début de l'entraînement** afin de se préparer calmement et méthodiquement.

L'entraînement débute exactement à l'heure prévue.

Pour profiter au maximum du temps qui vous est imparti et pour que l'entraînement puisse se dérouler dans les meilleures conditions :

- une concentration totale sera exigée.
- il aura lieu de s'arrêter immédiatement aux injonctions de l'entraîneur.

#### Matchs

La participation à tous les matchs est **indispensable**.

Le joueur qui pour une raison sérieuse, n'est pas en mesure de participer à un match devra en avertir son coach le plus rapidement possible.

Sachez que votre sélection repose sur divers facteurs dont notamment :

- la régularité aux entraînements
- la ponctualité,
- le travail fourni aux entraînements (la volonté de progresser).
- la forme du moment
- la tactique employée.
- la discipline
- la motivation
- l'adversaire

A l'issue de l'ultime entraînement de la semaine, les joueurs seront informés de la sélection du week-end. Les absents sont priés de prendre contact par téléphone avec leur coach pour connaître les décisions prises à leur sujet. Quelles que soit celles-ci, leur présence reste **indispensable**.

Veillez à toujours rester correct et fair-play envers les arbitres, les adversaires, les membres de votre club et des clubs adverses, le public, vos équipiers, ... aussi bien qu'pendant ou après le match ou l'entraînement.

#### Préparation d'avant match

La présence des joueurs sur le lieu de la rencontre est exigée **45 MINUTES avant le début du match**. Les joueurs seront en tenue pour l'échauffement AU MOINS 30 MINUTES avant l'heure du début du match. Cette demi-heure sera nécessaire pour le briefing et l'échauffement. Dès que l'arbitre siffle la « 1 minute », tous les joueurs viennent rejoindre le coach pour mettre au point les dernières détails.

Ensuite les cinq joueurs désignés pour débuter la rencontre se préparent tandis que les autres prennent place sur le banc. Sur le banc doivent uniquement figurer les joueurs réservistes et non repris sur la feuille de match, le coach, le parrain ou la marraine et l'éventuel assistant-coach, s'il y en a un.

Personne d'autre ne sera admis sur la banc ou derrière celui-ci !!!

### Règlement de la salle J Laloux (parquet)

L'accès au parquet est autorisé aux seuls joueur(s), coaches et délégués responsables chaussés de **basket** ou porteurs de chaussons.

Les chaussures de jogging sont strictement interdites.

Les chaussures sont disponibles au bar.

L'accès au parquet est formellement interdit à toute personne venant de l'extérieur en chaussures de basket et désirent marcher sur le parquet avec ces mêmes chaussures. Il est donc impératif de venir à la salle en souliers et de chausser des baskets propres.

Toutes les autres personnes doivent **obligatoirement** se diriger à l'étage réservé aux spectateurs ou à la cafétéria.

L'accès à la salle ainsi qu'à l'étage réservé aux spectateurs est interdit à toute personne en possession de nourriture ou de boisson(s).

L'usage des bouteilles d'eau pour les joueurs est interdit dans la salle; l'usage des bidons-gourdes est **obligatoire**.

La clef de vestiaire ainsi que les gourdes pour les visiteurs sont disponibles au bar sur simple demande.

#### Règlement d'ordre intérieur

#### Règlement de la salle J Laloux

#### Règlement interne du BC Avis à tous les membres du club

#### Cotisation

#### Déclaration d'accident

#### Certificat médical

#### Inscription dans une équipe

#### Etre parent au RBC Ciney

*Design de la page des entraîneurs*

**Découvrez nos entraîneurs**  
Ils forment vos jeunes dès le plus jeune âge.  
Ils transmettent leur passion du sport et leurs valeurs.  
Ils prônent une éducation basé sur le fair-play et le respect.

Martin Legros  
Équipe : U7 mixte  
0474 / 62 73 44  
martinlegros13@hotmail.com

Martin Legros  
Équipe : U7 mixte  
0474 / 62 73 44  
martinlegros13@hotmail.com

Martin Legros  
Équipe : U7 mixte  
0474 / 62 73 44  
martinlegros13@hotmail.com

Martin Legros  
Équipe : U7 mixte  
0474 / 62 73 44  
martinlegros13@hotmail.com

Martin Legros  
Équipe : U7 mixte  
0474 / 62 73 44  
martinlegros13@hotmail.com

Martin Legros  
Équipe : U7 mixte  
0474 / 62 73 44  
martinlegros13@hotmail.com

Martin Legros  
Équipe : U7 mixte  
0474 / 62 73 44  
martinlegros13@hotmail.com

Martin Legros  
Équipe : U7 mixte  
0474 / 62 73 44  
martinlegros13@hotmail.com

Nous trouver Nous contacter Nous suivre  
© Royal Basket Club de Ciney 2016 - Site réalisé par Mathieu Claessens

*Design de la page de coaching*

**Pour vous les coachs**  
Déposez ici des liens vers des sites externes et des fichiers à télécharger que vous veulez partager exclusivement avec tous les coachs du RBC Ciney.

**Site d'échauffements**  
Voici un site internet vraiment intéressant sur les échauffements.  
Mathieu Claessens  
[Accès au site](#)

**Feuille d'entraînement**  
J'ai fait une feuille de préparation d'entraînement à l'effigie du club, si ça intéresse quelqu'un...  
Mathieu Claessens  
[Télécharger le fichier](#)

**Tactique offensive**  
J'ai trouvé un site pas mal fait, avec de bonnes idées. J'ai mis aussi le PDF d'une phase qui me semble vraiment intéressante.  
Mathieu Claessens  
[Accès au site](#) [Télécharger le fichier](#)

**Partagez ici**  
Titre :  
Description : La description ici  
URL : URL du site à partager [Ajouter un fichier](#)  
[Partager](#)

Nous trouver Nous contacter Nous suivre  
© Royal Basket Club de Ciney 2016 - Site réalisé par Mathieu Claessens

*Design de la page d'un événement*

**Christmas basket 2016**  
Le RBC Ciney a le plaisir d'accueillir le plus grand stage d'initiation et de perfectionnement en Basket Ball de jeune en europe !

**Quand**  
Du 26/12/16 au 30/12/16

**Où**  
Au RBC Ciney

**Réservation**  
Par mail : basket.ciney@skynet.be  
Par téléphone : 0000 / 00 00 00  
[Réserver en ligne](#)

Nous trouver Nous contacter Nous suivre  
© Royal Basket Club de Ciney 2016 - Site réalisé par Mathieu Claessens

*Design de la page des téléchargements*

**Quelques fichiers à télécharger**  
Sur cette page vous trouverez quelques fichiers administratif à télécharger en format PDF

Déclaration d'accidents 2016 - 2017 <a href="#">Télécharger</a>	Remboursement mutuelle - Solidaris <a href="#">Télécharger</a>
Certificat médical 2016 - 2017 <a href="#">Télécharger</a>	Remboursement mutuelle - Chrétienne <a href="#">Télécharger</a>
Formulaire de réinscription 2016 - 2017 <a href="#">Télécharger</a>	Remboursement mutuelle - Libérale <a href="#">Télécharger</a>

Nous trouver Nous contacter Nous suivre  
© Royal Basket Club de Ciney 2016 - Site réalisé par Mathieu Claessens

*Design de la page des albums*

Accueil Club Calendrier Multimédia Complexe Contact Coaching Connection  
Albums photos

Découvrez tous nos albums photos

Saison 2016 -2017

Présentation des équipes	P1H - 09/12/16	R2D - 08/12/16	R2H - 08/12/16
U14G Prov A - 01/12/16	U14G Reg A - 01/12/16	UBF - 02/12/16	UBG - 02/12/16
P3H - 26/11/16	P1H - 26/11/16	R1D - 27/11/16	UT - 27/11/16

Saison 2015 -2016

Présentation des équipes	P1H - 09/12/16	R2D - 08/12/16	R2H - 08/12/16
U14G Prov A - 01/12/16	U14G Reg A - 01/12/16	UBF - 02/12/16	UBG - 02/12/16
P3H - 26/11/16	P1H - 26/11/16	R1D - 27/11/16	UT - 27/11/16

Nous trouver Nous contacter Nous suivre

© Royal Basket Club de Ciney 2016 - Site réalisé par Mathieu Claessens

*Design de la page d'un album*

Accueil Club Calendrier Multimédia Complexe Contact Coaching Connection  
Albums photos P1H - 02/11/16

P1 Homme - 02/11/16

Découvrez ici les photos prise lors de la rencontre du 02 novembre 2016 de l'équipe P1 Homme de Ciney contre celle de Belgrade.


Nous trouver Nous contacter Nous suivre

© Royal Basket Club de Ciney 2016 - Site réalisé par Mathieu Claessens

*Design de la page de profil*

Accueil Club Calendrier Multimédia Complexe Contact Coaching Profil Déconnexion  
Profil

Votre profil

Martin Legros

Informations personnelles

Date de naissance	28 / 09 / 1995
Lieu de naissance	Namur
Situation sociale	Célibataire
Adresse	Rue Concorde, 21 - 5590 CINEY
E-mail	martinlegros13@hotmail.com
Téléphone	0474 / 62 73 44
Famille	Legros

Nous trouver Nous contacter Nous suivre

© Royal Basket Club de Ciney 2016 - Site réalisé par Mathieu Claessens

*Design de la page de modification de profil*

Accueil Club Calendrier Multimédia Complexe Contact Coaching Profil Déconnexion  
Profil

Votre profil

Martin Legros

Informations personnelles

Date de naissance	28/09/1995
Lieu de naissance	Namur
Situation sociale	Célibataire
Adresse	Rue Concorde, 21 - 5590 Ciney
E-mail	martinlegros13@hotmail.com
Téléphone	0474/62 73 44
Famille	Legros

Annuler Enregistrer

Nous trouver Nous contacter Nous suivre

© Royal Basket Club de Ciney 2016 - Site réalisé par Mathieu Claessens

*Design de la page d'une équipe*

**Équipe Provinciale 1 Homme**

**Effectif**

Coach	Yannick Demoitte
Assistant	Guy Larache
Joueurs	Numéros
BALTHAZAR Yohan	14
BORREMANNS Allan	13
BRAET Martin	7
BRICHON Jordan	-
CLAESSENS Mathieu	8
DENEIL Jean-Sébastien	-
DEUXANT Julien	12
DEWEZ Clément	9
HAUTOT Sébastien	4
HENIN Esteban	6
LEGROS Martin	13
LEVAQUE Simon	10
PIROLAT Francois	15
SAMBON Maxime	5

**Horaires d'entraînement**

Jours	Heures	Salles
Mercredi	20h30	Petite salle
Vendredi	21h00	Grande salle

**L'équipe en photos**

**Nous trouver** **Nous contacter** **Nous suivre**

© Royal Basket Club de Ciney 2016 - Site réalisé par Mathieu Claessens

*Design de la page du complexe sportif*

**Notre complexe**

**Le trouver**

**Coordonnées**

**Plan d'accès**

**Comment m'y rendre**

**Nous contacter**

**Le louer**

**Tarif des locations**

Location à l'heure	Prix
Petite salle	15€
Grande salle	20€
Cafétéria	25€

**Location forfaitaire (+/- journée)**

Location	Prix
Petite salle	150€
Grande salle	200€
Cafétéria	250€
Cafétéria (entraînement)	130€
Cafétéria + Cuisine (salleuse comprise)	400€
Cafétéria + cuisine + petite salle	550€
Cafétéria et cuisine pour soirée privée	600€

**Comment ça marche**

Vous organisez un anniversaire, un souper, un stage de sport? Nous mettons nos infrastructures à votre disposition.

**Réservations :** basket.ciney@skynet.be

La location est à payer afin de **confirmer la réservation**. La **consommation d'énergie** est facturée au prix du jour/ KW (pour location à la journée).

**Caution de 200€** à payer lors de la confirmation de la réservation.

Le **nettoyage** est à charge du locataire. Si le nettoyage est mal fait, il y aura une amende de 150€.

**Réserver**

**Nous trouver** **Nous contacter** **Nous suivre**

© Royal Basket Club de Ciney 2016 - Site réalisé par Mathieu Claessens

*Design de la page de contact*

**Nous contacter**

**Vous avez oublié d'indiquer votre nom.**

**Il faut indiquer une adresse e-mail valide.**

**E-mail** Votre e-mail ici

**Sujet** Le sujet du message ici

**Message**  
Votre message ici

**Envoyer**

Votre message a bien été envoyé!

**Passer une réservation, besoin d'un renseignement, une idée à partager ?**

N'hésitez pas et envoyez un message via ce formulaire de contact ou via cette adresse mail : basket.ciney@skynet.be

**Nous trouver** **Nous contacter** **Nous suivre**

© Royal Basket Club de Ciney 2016 - Site réalisé par Mathieu Claessens

Lors du jury de janvier j'ai eu l'occasion de montrer mon travail à quelques professionnels du web, ils m'ont tous fait des remarques intéressantes, pertinentes et constructives. Ils m'ont montré que mon design devait encore être amélioré surtout en ce qui concerne la hiérarchie des éléments présents dans mes pages.

## Il y a encore du travail

Lors du jury, j'ai eu la chance de pouvoir présenter mes travaux à Bernard et Julien, de « Exit Studio », chez qui j'allais commencer mon stage quelques semaines plus tard. Ils m'ont proposé de revoir le design de mon projet au cours de mon stage. Une offre très aimable de leur part sachant que ce n'est pas du tout dans leurs obligations de maître de stage. J'ai bien entendu sauté sur cette occasion de bénéficier régulièrement de leurs remarques et conseils. Nous avons pris 30 minutes à 1 heure en fin de semaine pour évaluer l'avancement de mes travaux.

Grâce à ces séances de révision mon design a beaucoup évolué. Les conseils avisés de Julien et Bernard m'ont été très précieux. La hiérarchie de mes éléments s'est petit à petit améliorée, mes pages sont devenues de plus en plus claires et faciles à parcourir.

### Aperçu de l'évolution du design :

*Evolution du design de la page de contact*

The image shows a side-by-side comparison of the Royal Basket Club de Ciney's contact page design, illustrating its evolution from January to July 2017. On the left is the initial version from January, featuring a map of Ciney, a large 'Contactez-nous' button, and several input fields for name, email, subject, and message. A validation error for the email field is visible. On the right is the final version from July, which has been simplified and modernized. It includes a larger 'Envoyer' button, a success message for the sent message, and a footer with social media links and copyright information.

Réalisation du site du Royal Basket Club de Ciney en Laravel

## *Evolution du design de la page calendrier*

[Accueil](#) [Club](#) [Calendrier](#) [Multimédia](#) [Complexe](#) [Contact](#) [Coaching](#)

[Se connecter](#)

## Prochains matchs

Samedi 04 mars 2017

P1 Hommes	R2 Hommes	Filtres
<b>15h00</b> Rendez-vous 14h15 <b>RBC Ciney</b> Boulevard de la Bressure 6 4000 Liège <b>Belfius Namur</b> Capital A	<b>15h00</b> Départ à 14h15 <b>RBC Ciney</b> Boulevard de la Bressure 6 4000 Liège <b>Belfius Namur</b> Capital A	<a href="#">Équipes</a> <a href="#">R1 Dames</a> <a href="#">R2 Homme</a> <a href="#">R2 Dames</a> <a href="#">P1 Dames</a> <a href="#">P1 Hommes</a> <a href="#">P2 Dames</a> <a href="#">P3 Hommes</a> <a href="#">U18 Garçons Prov A</a>
<b>R1 Dames</b>	<b>P1 Dames</b>	
<b>15h00</b> Départ à 14h15 <b>RBC Ciney</b> Boulevard de la Bressure 6 4000 Liège <b>Belfius Namur</b> Capital A	<b>15h00</b> Départ à 14h15 <b>RBC Ciney</b> Boulevard de la Bressure 6 4000 Liège <b>Belfius Namur</b> Capital A	
<b>P3 Hommes</b>	<b>U18 Garçons Prov A</b>	
<b>15h00</b> Départ à 14h15 <b>RBC Ciney</b> Boulevard de la Bressure 6 4000 Liège <b>Belfius Namur</b> Capital A	<b>15h00</b> Départ à 14h15 <b>RBC Ciney</b> Boulevard de la Bressure 6 4000 Liège <b>Belfius Namur</b> Capital A	
<a href="#">PLUS DE MATCHS</a>		

### Résultats

Samedi 04 mars 2017				Filtres
P1 DAMES	R2 HOMMES	R2 DAMES	P1 HOMMES	Équipes
RBC Ciney <b>125</b>	Bellus Namur Capital A <b>123</b>	RBC Ciney <b>125</b>	Bellus Namur Capital A <b>123</b>	R1 Damas
RBC Ciney <b>87</b>	Bellus Namur Capital A <b>102</b>	RBC Ciney <b>87</b>	Bellus Namur Capital A <b>102</b>	R2 Homme
U18 GARÇONS PROV A		P3 HOMMES		R2 Damas
RBC Ciney <b>125</b>	Bellus Namur Capital A <b>123</b>	RBC Ciney <b>125</b>	Bellus Namur Capital A <b>123</b>	P1 Damas
PLUS DE RÉSULTATS				P1 Hommes
				P2 Damas
				P3 Hommes
				U18 Garçons Prov A

Samedi 04 mars 2017				Résultats
P1 DAMES		P2 HOMMES		Équipes
RBC Ciney	Bellfus Namur Capital A	RBC Ciney	Bellfus Namur Capital A	D2 Dames
<b>125</b>	<b>123</b>	<b>125</b>	<b>123</b>	R1 Dames
R2 DAMES		P1 HOMMES		R2 Homme
RBC Ciney	Bellfus Namur Capital A	RBC Ciney	Bellfus Namur Capital A	R2 Dames
<b>87</b>	<b>102</b>	<b>87</b>	<b>102</b>	P1 Dames
U18 GARÇONS PROV A		P3 HOMMES		P1 Hommes
RBC Ciney	Bellfus Namur Capital A	RBC Ciney	Bellfus Namur Capital A	P2 Dames
<b>125</b>	<b>123</b>	<b>125</b>	<b>123</b>	P3 Hommes
PLUS DE RÉSULTATS				U18 Garçons Prov A

# Festivités

# Activités du club

*Evolution du design de la page d'accueil*

The image shows two versions of the Royal Basket Club de Ciney website side-by-side. The left version is from 2016 and the right version is from 2017. Both versions feature a large banner at the top showing a group photo of the basketball team. The 2016 version has a more cluttered layout with multiple sections like 'Matchs de la semaine', 'Dernière news', and 'Actu facebook'. The 2017 version is more organized, featuring a prominent 'Matches à venir' section for the next game on Saturday, March 4, 2017, and a cleaner 'Actualité du club' section with news items and images.

Les remarques et les conseils que Julien et Bernard ont formulés tout au long du stage étaient pertinents et constructifs. Ils ont permis de m'améliorer en terme de design en dehors des projets qu'ils m'avaient donnés pour le stage proprement dit.

Un processus qui m'a pris beaucoup de temps, peut-être un peu trop, mais qui m'a permis de mieux comprendre ce qu'est un bon design et de combler les lacunes que j'avais en design. J'ai compris comment mettre en valeur les informations importantes dans une page, laisser de l'espace entre les éléments et présenter un design léger.

## On en voit le bout

Le chemin fut long mais a fini par aboutir sur un design dont je suis fier. On peut voir un net progrès par rapport au design que j'avais présenté au jury de janvier et sur la globalité de mes travaux.

### Aperçu du design final :

*Design final de la page d'accueil*

The screenshot shows the main homepage of the RBC Ciney website. At the top is a navigation bar with links for Accueil, Club, Calendrier, Multimédia, Complexe, Contact, Coaching, and Se connecter. Below the navigation is a large banner featuring a group photo of the basketball team. To the left of the banner is a sidebar with links for Conseil d'administration, Entraineurs, Équipes, and Réglements. The main content area has a heading "Matchs à venir" (Upcoming Matches) with a grid of match details. Each row contains a team name, date, time, location, and opponent. Below the grid is a "Filtrer par équipes" (Filter by teams) dropdown menu with checkboxes for R1 Dames, R2 Hommes, P1 Hommes, P3 Hommes, and U18 Garçons Prov A. At the bottom of the page are buttons for "VOIR TOUS NOS MATCHS" and "VOIR NOS RÉSULTATS".

*Suite du design final de la page d'accueil*

This screenshot shows a promotional section for the "Christmas BASKET" event. It features several images: a person pouring cheese onto a plate, a basketball player, a group of people cheering, and a close-up of a raclette dish. There is also a calendar for March 2017 with specific dates highlighted. Below the images is a section titled "Nos partenaires" (Our partners) showing logos for Adeps. Further down is a section titled "Nos derniers albums photo" (Our latest photo albums) displaying thumbnail images of team photos from various matches.

Design final de la page du calendrier

**Prochains matches**

P1 Hommes	P1 Hommes
Sam 04 mars <b>15h00</b> Rendez-vous 14h15	RBC Ciney Rue de la brasserie 6 4000 Liège Bellus Namur Capital A
Sam 04 mars <b>17h00</b> Rendez-vous 14h15	RBC Ciney Rue de la brasserie 6 4000 Liège Bellus Namur Capital A
<b>R1 Dames</b>	<b>R2 Hommes</b>
Sam 04 mars <b>19h00</b> Rendez-vous 14h15	RBC Ciney Rue de la brasserie 6 4000 Liège Bellus Namur Capital A
Dim 05 mars <b>17h00</b> Rendez-vous 14h15	RBC Ciney Rue de la brasserie 6 4000 Liège Bellus Namur Capital A
<b>R2 Hommes</b>	<b>R1 Dames</b>
Dim 05 mars <b>17h00</b> Rendez-vous 14h15	RBC Ciney Rue de la brasserie 6 4000 Liège Bellus Namur Capital A
<b>MATCHES SUIVANT</b>	
<a href="#">Filtrer par équipes</a> <input checked="" type="checkbox"/> R1 Dames <input checked="" type="checkbox"/> R2 Homme <input type="checkbox"/> R2 Dames <input type="checkbox"/> P1 Dames <input checked="" type="checkbox"/> P1 Hommes	

**Résultats**

P1 DAMES	P1 DAMES
Samedi 04 mars 2017 RBC Ciney 125 Bellus Namur Capital A 123	Samedi 04 mars 2017 RBC Ciney 125 Bellus Namur Capital A 123
<b>P1 DAMES</b>	<b>P1 DAMES</b>
Samedi 04 mars 2017 RBC Ciney 125 Bellus Namur Capital A 123	Samedi 04 mars 2017 RBC Ciney 125 Bellus Namur Capital A 123
<b>P1 DAMES</b>	<b>P1 DAMES</b>
Samedi 04 mars 2017 RBC Ciney 125 Bellus Namur Capital A 123	Samedi 04 mars 2017 RBC Ciney 125 Bellus Namur Capital A 123
<b>PLUS DE RÉSULTATS</b>	
<a href="#">Filtrer par équipes</a> <input checked="" type="checkbox"/> R1 Dames <input checked="" type="checkbox"/> R2 Homme <input type="checkbox"/> R2 Dames <input type="checkbox"/> P1 Dames	

**Activités du club**

05 mars	Souper de présentation d'équipes	05 mars	Souper de présentation d'équipes
Service assuré par U18 Garçon Prov A - P1 Hommes	Service assuré par U18 Garçon Prov A - P1 Hommes	Service assuré par U18 Garçon Prov A - P1 Hommes	Service assuré par U18 Garçon Prov A - P1 Hommes
<b>05 mars</b>	<b>Souper de présentation d'équipes</b>	<b>05 mars</b>	<b>Souper de présentation d'équipes</b>
Service assuré par U18 Garçon Prov A - P1 Hommes	Service assuré par U18 Garçon Prov A - P1 Hommes	Service assuré par U18 Garçon Prov A - P1 Hommes	Service assuré par U18 Garçon Prov A - P1 Hommes
<b>05 mars</b>	<b>Souper de présentation d'équipes</b>	<b>05 mars</b>	<b>Souper de présentation d'équipes</b>
Service assuré par U18 Garçon Prov A - P1 Hommes	Service assuré par U18 Garçon Prov A - P1 Hommes	Service assuré par U18 Garçon Prov A - P1 Hommes	Service assuré par U18 Garçon Prov A - P1 Hommes
<b>05 mars</b>	<b>Souper de présentation d'équipes</b>	<b>05 mars</b>	<b>Souper de présentation d'équipes</b>
Service assuré par U18 Garçon Prov A - P1 Hommes	Service assuré par U18 Garçon Prov A - P1 Hommes	Service assuré par U18 Garçon Prov A - P1 Hommes	Service assuré par U18 Garçon Prov A - P1 Hommes
<b>05 mars</b>	<b>Souper de présentation d'équipes</b>	<b>05 mars</b>	<b>Souper de présentation d'équipes</b>
Service assuré par U18 Garçon Prov A - P1 Hommes	Service assuré par U18 Garçon Prov A - P1 Hommes	Service assuré par U18 Garçon Prov A - P1 Hommes	Service assuré par U18 Garçon Prov A - P1 Hommes
<b>PLUS D'ACTIVITÉS</b>			

Rue Saint-Quentin, 10b  
5590 Ciney  
© Royal Basket Club de Ciney 2017 | Site réalisé par Mathieu Claessens

basket.ciney@skynet.be  
Nous suivre

Design final de la page des règlements

**Nos règles**

**Règlement d'ordre intérieur**

**Introduction**

Vous trouverez ci-dessous le règlement de discipline sportive du club. Une première lecture vous laissera sans doute une impression de régime sévère, de « dictature »... « Par la suite, vous serez toutefois « obligés » d'admettre que tout ce qui y figure est logique et nécessaire au bon fonctionnement d'un club qui se veut organisé.

Quoiqu'il en soit, nous insistons sur le fait que ce règlement sera de stricte application en toutes circonstances.

**Entraînements**

La participation aux entraînements est indispensable.

Le joueur qui pour une raison sérieuse (cas de force majeure, maladie, blessure, travail,...) n'est pas en mesure de participer à un entraînement devra en informer son entraîneur au plus tard 2 heures avant le début de celui-ci. Il est vivement conseillé d'arriver 1/4 d'heure avant le début de l'entraînement afin de se préparer calmement et méthodiquement.

L'entraînement débutera exactement à l'heure prévue.

Pour profiter au maximum du temps qui vous est imparti et pour que l'entraînement puisse se dérouler dans les meilleures conditions :

- une concentration totale sera exigée,
- il y aura lieu de s'arrêter immédiatement aux injonctions de l'entraîneur.

**Matches**

La participation à tous les matchs est indispensable.

Le joueur qui pour une raison sérieuse, n'est pas en mesure de participer à un match devra en avertir son coach le plus rapidement possible.

Sachez que votre sélection repose sur divers facteurs dont notamment :

- la régularité aux entraînements
- la ponctualité
- le travail fourni aux entraînements (la volonté de progresser)
- la forme du moment
- la tactique employée
- la discipline
- la motivation
- l'adversaire

À l'issue de l'ultime entraînement de la semaine, les joueurs seront informés de la sélection du week-end. Les absents sont priés de prendre contact par téléphone avec leur coach pour connaître les décisions prises à leur sujet. Quelles que soit celles-ci, leur présence reste indispensable.

Veillez à toujours rester correct et fair-play envers les arbitres, les adversaires, les membres de votre club et des clubs adverses, le public, vos équipiers,... aussi bien avant que pendant ou après le match ou l'entraînement.

**Préparation d'avant match**

La présence des joueurs sur le lieu de la rencontre est exigée **45 MINUTES avant le début du match**. Les joueurs seront en tenue pour l'échauffement AU MOINS 30 MINUTES avant l'heure du début du match. Cette demi-heure sera nécessaire pour le briefing et l'échauffement. Dès que l'arbitre siffle la « 1 » = minute, tous les joueurs viennent rejoindre le coach pour mettre au point les derniers détails.

Ensuite les cinq joueurs désignés pour débuter la rencontre se préparent tandis que les autres prennent place sur le banc. Sur le banc doivent uniquement figurer les joueurs réservistes et non repris sur la feuille de match, le coach, le parrain ou la marraine et l'éventuel assistant-coach, s'il y en a un.

Personne d'autre ne sera admis sur le banc ou derrière celui-ci !!!

**Règlement de la salle J Laloux (parquet)**

L'accès au parquet est autorisé aux seuls joueur(s), coaches et délégués responsables chaussés de basket ou porteurs de chaussures.

Les chaussures de jogging sont strictement interdites.

Les chaussures sont disponibles au bar.

L'accès au parquet est formellement interdit à toute personne venant de l'extérieur en chaussures de basket et désirant marcher sur le parquet avec ces mêmes chaussures. Il est donc impératif de venir à la salle en souliers et de chaussier des baskets propres.

Toutes les autres personnes doivent **obligatoirement** se diriger à l'étage réservé aux spectateurs ou à la cafétéria.

L'accès à la salle ainsi qu'à l'étage réservé aux spectateurs est interdit à toute personne en possession de nourriture ou de boissons).

L'usage des bouteilles d'eau pour les joueurs est interdit dans la salle; l'usage des bidons-gourdes est **obligatoire**.

La clé du vestiaire ainsi que les gourdes pour les visiteurs sont disponibles au bar sur simple demande.

**Règlement interne du BC**

Par le simple fait de leur affiliation au RBC Ciney, tous les membres sont tenus de respecter le règlement de discipline sportive (charte du joueur) et les décisions du Conseil d'Administration sous peine de sanction voire d'exclusion du club.

De même, par le simple fait de son inscription au club, chaque joueur(se) ou son représentant légal est tenu(e) de participer aux manifestations extra-sportives, (foires,...) et de respecter les consignes et les tours de rôles concernant les différentes activités du club.

Rue Saint-Quentin, 10b  
5590 Ciney  
© Royal Basket Club de Ciney 2017 | Site réalisé par Mathieu Claessens

basket.ciney@skynet.be  
Nous suivre

Design final de la page d'une équipe

**Équipe Provinciale 1 Hommes**

**Prochains matches**

Samedi 05 mars	Samedi 05 mars
<b>15h00</b> Rendez-vous 14h15	<b>RBC Ciney</b> Rue de la brasserie 6 4000 Liège Bellfus Namur Capital A
<b>15h00</b> Rendez-vous 14h15	<b>RBC Ciney</b> Rue de la brasserie 6 4000 Liège Bellfus Namur Capital A
<b>15h00</b> Rendez-vous 14h15	<b>RBC Ciney</b> Rue de la brasserie 6 4000 Liège Bellfus Namur Capital A
<b>15h00</b> Rendez-vous 14h15	<b>RBC Ciney</b> Rue de la brasserie 6 4000 Liège Bellfus Namur Capital A

**Résultats**

ANCIENS RÉSULTATS	
<b>DIMANCHE 12 MARS</b>	
<b>RBC Ciney</b>	<b>Bellfus Namur Capital A</b>
<b>125</b>	<b>123</b>
<b>DIMANCHE 05 MARS</b>	
<b>RBC Ciney</b>	<b>Bellfus Namur Capital A</b>
<b>87</b>	<b>102</b>
<b>SAMEDI 04 MARS</b>	
<b>RBC Ciney</b>	<b>Bellfus Namur Capital A</b>
<b>125</b>	<b>123</b>

**Effectif**

Entraîneur		Joueurs	
	Martin Legros 0474 / 627344 martinlegros13@hotmail.com	<b>BALTHAZAR Yohan</b>	N° 14
	Martin Legros 0474 / 627344 martinlegros13@hotmail.com	<b>BALTHAZAR Yohan</b>	N° 14
	Martin Legros 0474 / 627344 martinlegros13@hotmail.com	<b>BALTHAZAR Yohan</b>	N° 14
<b>Assistant</b>	<a href="#">PLUS D'INFOS &gt;</a>	<b>BALTHAZAR Yohan</b>	<a href="#">PLUS D'INFO &gt;</a>
<b>Entrainement</b>		<b>BALTHAZAR Yohan</b>	N° 14
Jour	Heure	Salles	
Mercredi	20h30	Petite salle	
Vendredi	21h00	Grande salle	

**L'équipe en images**

R1D - 04 mars 2017		
<a href="#">VOIR L'ALBUM PHOTO &gt;</a>	<a href="#">VOIR L'ALBUM PHOTO &gt;</a>	<a href="#">VOIR L'ALBUM PHOTO &gt;</a>
R1D - 04 mars 2017		
<a href="#">VOIR L'ALBUM PHOTO &gt;</a>	<a href="#">VOIR L'ALBUM PHOTO &gt;</a>	<a href="#">VOIR L'ALBUM PHOTO &gt;</a>

Design final de la page de contact

**Contactez-nous**

Votre nom ici  
Votre adresse mail ici  
Le sujet du message ici  
Votre message ici

Votre message à bien été envoyé !

**ENVOYER**

**RBC Ciney**  
Rue Saint-Quentin 10  
Agrandie le plan

**Rue Saint-Quentin 10**  
Centre Culturel de Ciney

**basket.ciney@skynet.be**  
[Nous suivre](#)

© Royale Basket Club de Ciney 2017 | Site réalisé par Mathieu Claessens

Design final de la page du complexe sportif

**Notre complexe sportif**

**Location à l'heure**

Petite salle	15 €
Grande salle	20 €
Cafétaria	25 €

**Location à la journée**

Petite salle	150 €
Grande salle	200 €
Cafétaria	250 €
Cafétaria (enterrement)	130 €
Cafétaria + Cuisine (vaisselle comprise)	400 €
Cafétaria + Cuisine + Petite salle	550 €
Cafétaria + Cuisine (soirée privée)	600 €

Vous organisez un anniversaire, un souper, un stage de sport ? Nous mettons nos infrastructures à votre disposition.

La location est à payer afin de confirmer la réservation.

La consommation d'énergie est facturée au prix du jour/KW (pour location à la journée).

Caution de 200€ à payer lors de la confirmation de la réservation.

Le nettoyage est à charge du locataire. Si le nettoyage est mal fait, il y aura une amende de 150€.

Reservation : [basket.ciney@skynet.be](mailto:basket.ciney@skynet.be)

**RÉSERVEZ DÉS MAINTENANT**

**RBC Ciney**  
Rue Saint-Quentin 10  
Agrandie le plan

**Rue Saint-Quentin 10**  
Centre Culturel de Ciney

**basket.ciney@skynet.be**  
[Nous suivre](#)

© Royale Basket Club de Ciney 2017 | Site réalisé par Mathieu Claessens

*Design final de la page des entraîneurs*

The page displays a grid of 10 coach profiles, each featuring a small photo, the coach's name, their team (U8 Garçons or U9 Filles), and their contact details (phone number and email). A 'PLUS D'IMOS' button is visible at the bottom right of the grid.

*Design final de la page de coaching*

The page lists several training session preparation sheets, each with a title, a brief description, the date, the author, and two buttons: 'ACCÈS AU SITE' and 'TÉLÉCHARGER'. A 'Fiche de préparation' section is also shown.

*Design final de la page d'un événement*

The page details a basketball stage for young people in Europe, held from December 26 to December 30, 2016, in Ciney. It includes a large promotional image, registration information, and contact details for the RBC Ciney.

*Design final de la page de profil*

The profile page for Martin Legros shows his personal information (date of birth), contact details (telephone and email), family information (wife Véronique Legros), and his role in the club (Player-Trainer-Member C.A.). It also lists his coached teams (U18 Garçons Prov A).

*Design final de la page des albums*

The screenshot shows the 'Albums photos' section of the website. At the top, there's a navigation bar with links for Accueil, Club, Calendrier, Multimédia, Complexe, Contact, Coaching, and Se connecter. Below the navigation is a header 'Nos album photos' with a sub-header 'Saison 2016 - 2017'. A large grid of basketball team photos is displayed, each with a caption and a 'Voir l'album' link. The photos include group shots of men's and women's teams on the court.

*Design final de la page d'un album*

The screenshot shows a single album page for 'P1H - 04 mars 2017'. At the top, there's a header 'Albums photos' and a link 'P1H - 04 mars 2017'. Below the header, a descriptive text reads: 'Découvrez ici les photos prise lors de la rencontre du 02 novembre 2016 de l'équipe P1 Homme de Ciney contre celle de Belgrade.' A grid of 12 photos from the album is displayed, including individual shots and group shots of the basketball team.

*Design final de la page des téléchargements*

The screenshot shows the 'Fichier à télécharger' section of the website. At the top, there's a navigation bar with links for Accueil, Club, Calendrier, Multimédia, Complexe, Contact, Coaching, and Se connecter. Below the navigation is a header 'Fichier à télécharger'. A text block states: 'Sur cette page vous trouverez quelques fichiers administratif à télécharger en format PDF'. A table titled 'Liste des fichiers' shows several downloadable files, each with a 'Télécharger' button. The files include: Déclaration d'accidents 2016 - 2017, Certificat médical 2016 - 2017, Formulaire de réinscription 2016 - 2017, Remboursement mutuelle - Solidaris, Remboursement mutuelle - Chrétienne, and Remboursement mutuelle - Libérale.

The screenshot shows the footer area of the website. It includes the club's address (Rue Saint-Quentin, 10b, 5590 Ciney), email (basket.ciney@skynet.be), a 'Nous suivre' link, and copyright information ('© Royale Basket Club de Ciney 2017 | Site réalisé par Mathieu Claessens').

## Intégration

### Lara vèle

Une fois le design clôturé et validé par Jean-Paul je pouvais passer à l'intégration du site. Deux choix s'offraient alors à moi : construire le site en Wordpress ou en Laravel.

Je connaissais déjà bien Wordpress sachant qu'on avait déjà utilisé ce framework dans le cadre de ma formation, notre premier gros projet « sur un baobab », notre portfolio, ainsi que les deux projets de troisième année : « CSI Angleur » et « Espace P... ». Pour compléter ma formation en Wordpress j'ai effectué mon stage chez Exit Studio qui utilisait principalement ce CMS. J'avais donc une bonne connaissance de ce framework mais ce n'était pas celui qu'on me recommandait pour ce type de projet.

D'un autre coté il y avait Laravel, un framework que je ne connaissais pas, ou très peu. Mr. Vilain nous l'avait très rapidement introduit sans vraiment nous donner de projet entier à réaliser. C'était donc se lancer dans l'inconnu sur le plus gros projet de ma formation, un risque certain mais un beau challenge à relever et un sacré atout à ajouter à mon panel de compétences.

J'ai longtemps hésité entre les deux car l'une de mes plus grandes peurs grandissantes en me rapprochant de la vie professionnelle a toujours été de ne pas être assez autodidacte. Je sais depuis toujours que dans le milieu du web tout avance très vite et sans cesse, j'étais conscient qu'en tant que professionnel du web ma formation ne s'arrêterait pas une fois diplômé, j'allais devoir me mettre à jour sans arrêt et apprendre de nouvelles technologies tout au long de ma carrière. Je n'étais pas confiant en ma capacité d'apprendre par moi-même, j'ai donc décidé de relever le défi et de faire mon projet en Laravel. Non seulement pour me prouver que j'en étais capable mais pour rajouter une corde à mon arc par la même occasion.

### Un sac à dos bien rempli

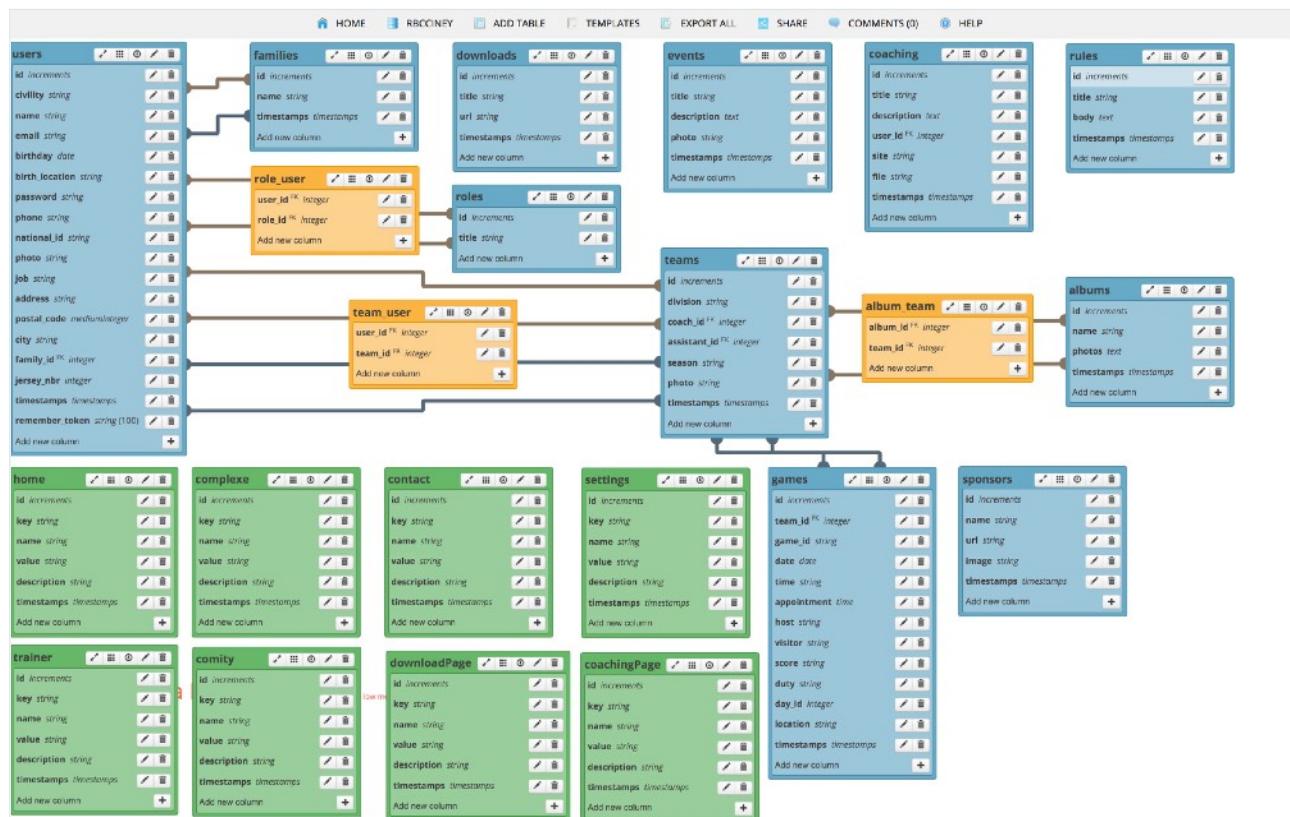
Une fois décidé à utiliser Laravel, je me suis attaqué à l'administration du site. Pour moi, un choix logique de travailler en premier sur la gestion du contenu avant de travailler sur le contenu lui-même. Cela me rendrait la vie plus simple ensuite pour modifier les contenus et les mettre en forme.

Contrairement à Wordpress, Laravel n'a pas d'administration intégrée au framework. Je devais donc partir de zéro. J'ai retroussé mes manches et je m'y suis mis sans attendre. J'ai commencé par coder tout par moi-même mais j'ai vite constaté que je fonçais dans le mur et que cela me prendrait une éternité en voyant toutes les fonctionnalités que j'avais à gérer. Après en avoir parlé à Mr. Vilain, il m'a conseillé d'utiliser Backpack, un des framework les plus populaires pour le panneau d'administration avec Laravel.

Ma première réaction a été de faire un projet test pour découvrir comment ce framework fonctionnait et rapidement le prendre en mains. Une fois chose faite je devais disposer de ma base de données pour pouvoir utiliser efficacement Backpack.

J'avais entendu parler d'un site « Laravel Schema designer » avec lequel je pouvais créer ma base de données simplement et exporter les fichiers de migration, seeding, modèle et contrôleur. Un outil très pratique, facile à prendre en mains et qui m'a permis de gagner beaucoup de temps à la création de ma base de données. Par ailleurs, une fois dans le projet il m'était plus facile d'effectuer mes modifications directement dans les fichiers plutôt que de repasser par le site. Un outil très utile pour lancer un projet mais pas si pratique lorsqu'on a des retouches à faire en cours de production.

Visualisation du site « Laravel Schema Designer »



Une fois la base de données réalisée, j'étais enfin prêt à me lancer dans la confection de l'administration. La mise en place était assez simple et me fournissait déjà la plupart des fonctionnalités dont j'avais besoin (un système d'identification, un panel settings, un système de log, un système de gestion des fichiers et un système de backup). Les bases de l'administration étaient mises en place, je pouvais dès lors ajouter mes fonctionnalités spécifiques. J'avais une marche à suivre pour ajouter un panel d'administration qui se répétait pour chacun d'entre eux. Je vais donc vous en décrire un comme exemple sachant que c'était assez répétitif une fois que j'avais compris la marche à suivre.

La première étape était de créer une table dans ma base de données, ce qui était déjà fait au préalable. Dans cet exemple : la table « roles ». Une fois chose faite, il fallait que je crée un model lié à cette table « *Role.php* ». Ce fichier contient les dépendances de base.

#### Dépendances

```
1 <?php
2
3 namespace App\Models;
4
5 use Backpack\CRUD\CrudTrait;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\Builder;
```

Ensuite je devais configurer le modèle pour le lier à la bonne table dans ma base de données. Pour ce faire, je devais configurer quelques variables globales dont Backpack avait besoin. Ces informations sont à définir au début de notre class, la variable « \$table » pour le nom de la table lié à ce modèle, « \$primaryKey » pour la clé primaire de cette même table, « \$timestamps » est un boolean pour spécifier si on a besoin des colonnes « created\_at » et « updated\_at », et la dernière dont j'avais besoin, « \$fillable » pour définir les colonnes de la table que je pouvais enregistrer via mon administration. Dans cet exemple il n'y a qu'une colonne « title » car le lien entre les utilisateurs et les rôles sont d'un rapport « N à N » donc se font via une table d'association.

### Définition de nos variables globales

```

9  class Role extends Model
10 {
11     use CrudTrait;
12
13     /*
14     |-----|
15     | GLOBAL VARIABLES
16     |-----|
17     */
18
19     protected $table = 'roles';
20
21     protected $primaryKey = 'id';
22
23     public $timestamps = false;
24
25     // protected $guarded = ['id'];
26
27     protected $fillable = [ 'title' ];
28
29     // protected $hidden = [];
30
31     // protected $dates = [];

```

Une fois le modèle modifié et les variables définies, l'étape suivante était de créer le contrôleur du modèle que je venais de mettre en place, « RoleCrudController.php ». Comme pour le modèle, j'avais quelques dépendances de base vers le fichier source « CrudController.php » et les validations de formulaires.

### Dépendances de base

```

1  <?php
2
3  namespace App\Http\Controllers\Admin;
4
5  use Backpack\CRUD\app\Http\Controllers\CrudController;
6
7  // VALIDATION: change the requests to match your own file names if you need form validation
8  use App\Http\Requests\RoleCreateRequest as StoreRequest;
9  use App\Http\Requests\RoleUpdateRequest as UpdateRequest;
10

```

Ce fichier étant généré par le framework, je pouvais y retrouver trois fonctions prédéfinies : une fonction « setup », une fonction « store » et une fonction « update ». C'est dans la fonction « setup » que je définis les différents éléments que je veux voir apparaître dans mon panel d'administration des rôles. Dans un premier temps, je définissais les informations de base de Backpack, le modèle lié à ce contrôleur, la route que contrôle ce fichier et le nom de mes entités au sein de mes vues.

#### *Définition de mes informations de base*

```

11  class RoleCrudController extends CrudController
12 {
13     public function setup()
14     {
15         /*
16         | -----
17         |-----| BASIC CRUD INFORMATION
18         |-----|
19         |
20         */
21         $this->crud->setModel('App\Models\Role');
22
23         $this->crud->setRoute(config('backpack.base.route_prefix') . '/role');
24
25         $this->crud->setEntityNameStrings('un rôle', 'rôle');

```

Après ces informations de base, je devais définir les différents champs à remplir pour l'ajout ou la modification d'un élément dans ma table. Pour cela j'utilisais la fonction `addField()` en y spécifiant quelques paramètres tels que : « label » pour le titre du champ qui allait s'afficher, « name » pour indiquer à quelle colonne de ma table ce champ fait référence, « type » pour le type d'input que je voulais créer et « attributes » pour ajouter des attributs quelconques à l'input (un placeholder par exemple). Dans certains types de champ, comme un « select » qui pouvait faire référence à des éléments d'une autre table (par exemple des utilisateurs) je devais ajouter quelques attributs pour importer les éléments voulus d'une autre table : « model » pour le modèle lié à la table dont j'ai besoin, « entity » pour la méthode qui définit la relation entre les deux tables dans le modèle cité avant, « attribute » pour la clé étrangère au sein de ma table et « pivot » dans le cas où il y avait besoin d'une table d'association.

*Déclaration des champs*

```

31     // ----- CRUD FIELDS
32
33     $this->crud->addField(
34
35         [
36             [
37                 'label' => 'Titre *',
38                 'name' => 'title',
39                 'attributes' => [
40                     [
41                         'placeholder' => 'Joueur',
42                     ],
43                 ],
44                 'both'
45             );
46
47     $this->crud->addField(
48
49         [
50             [
51                 'label'      => 'Membres',
52                 'type'       => 'select2_multiple',
53                 'name'       => 'users', // the method that defines the relationship in your Model
54                 'entity'     => 'users', // the method that defines the relationship in your Model
55                 'attribute'  => 'name', // foreign key attribute that is shown to user
56                 'model'      => 'App\User', // foreign key model
57                 'pivot'      => true, // on create&update, do you need to add/delete pivot table entries?
58             ],
59         ],
60         'both'
61     );

```

Après ces paramètres je décidais ensuite si je voulais que ce champ soit visible dans ma vue de modification ou d'ajout d'élément ou les deux en entrant la valeur « update », « create » ou « both »

Après avoir défini les champs de formulaires, je définissais les différentes colonnes que je voulais afficher dans la vue récapitulative de mon panel d'administration. L'idée est la même que pour les champs avec une fonction un peu différente *addColumns()* dans laquelle je pouvais définir plusieurs colonnes à la fois. Vu que le framework était bien fait, les paramètres à définir pour les colonnes étaient similaires à ceux des champs respectif sdéfinis plus haut, ce qui rendait le travail un peu plus facile.

Le fichier se termine avec les deux autres fonctions (store et update) qui sont générées par Backpack et qui n'ont pas besoin d'être modifiées la plupart du temps.

### Définition des colonnes

```

55     // ----- CRUD COLUMNS
56
57     $this->crud->addColumn([
58         [
59             'label' => 'Titre',
60             'name' => 'title',
61         ],
62         [
63             'label'      => 'Membres', // Table column heading
64             'type'       => 'select_multiple',
65             'name'       => 'users', // the method that defines the relationship in your Model
66             'entity'     => 'users', // the method that defines the relationship in your Model
67             'attribute'  => 'name', // foreign key attribute that is shown to user
68             'model'      => "App\User", // foreign key model
69         ],
70     ]);

```

### Fonctions générées par backpack

```

72     public function store(StoreRequest $request)
73     {
74         // your additional operations before save here
75         $redirect_location = parent::storeCrud();
76
77         // your additional operations after save here
78
79         return $redirect_location;
80     }
81
82     public function update(UpdateRequest $request)
83     {
84         // your additional operations before save here
85         $redirect_location = parent::updateCrud();
86
87         // your additional operations after save here
88
89         return $redirect_location;
90     }

```

J'en ai fini avec le contrôleur pour le moment, je passe donc au fichier suivant que je dois modifier : « *RoleRequest.php* ». La première fonction, *authorize()*, dont je me sers a pour but d'autoriser l'accès aux pages concernant le contrôleur qui fait appel à cette classe. C'est donc ici que je gère les différents droits d'accès par rapport aux rôles attribués à l'utilisateur connecté. Dans ce cas, je n'autorise l'accès à ce panel d'administration que pour les utilisateurs ayant le rôle « Web Developer » ou « Web Master ».

La deuxième fonction que j'utilise, *rules()*, est celle où je vais définir les règles de validation des différents champs que j'ai créés dans mon modèle lors d'une étape antérieure.

#### Fonction d'autorisation au panel

```

15  public function authorize()
16  {
17      // only allow access if the user logged in is web master or developer
18      if( Auth::user()->hasRole( 'Web Developer' ) ) {
19          $authorised = true;
20      } elseif ( Auth::user()->hasRole( 'Web Master' ) ) {
21          $authorised = true;
22      } elseif ( Auth::user()->hasRole( 'Web Communication' ) ) {
23          $authorised = false;
24      } else {
25          $authorised = false;
26      }
27      return $authorised;
28  }

```

#### Fonction de validation des formulaires

```

35  public function rules()
36  {
37      return [
38          'title' => 'Required|max:50|Unique:roles',
39      ];
40  }

```

Dans certains cas, comme celui-ci j'avais besoin de règles différentes pour la fonction « store » et la fonction « update » c'est pourquoi j'ai créé les fichiers « *RoleCreateRequest.php* » et « *RoleUpdateRequest.php* », ils sont essentiellement les mêmes hormis la définition des règles de validation.

Une fois le fichier request terminé, il me reste à autoriser la route que mon panel d'administration utilise. Pour ça, je suis d'abord allé dans le fichier « *RouteServiceProvider.php* » pour y trouver la fonction *map()* et y ajouter les routes de mon administration Backpack, *mapAdminRoutes()*.

#### *Ajout des routes de mon administration*

```

36     public function map()
37     {
38         $this->mapApiRoutes();
39
39         // Register CRUD routes
40
40         $this->mapAdminRoutes();
41
41         $this->mapWebRoutes();
42     }

```

Cet ajout me permet de créer un fichier de route, *admin.php*, spécifique à mon administration et d'y ajouter toutes les routes nécessaire pour mes panneaux d'administration.

#### *Ajout de la route du panel dans mon fichier « admin.php »*

```

11
12     CRUD::resource('role', 'RoleCrudController');
13

```

Il ne me reste plus qu'à ajouter un lien dans ma vue qui pointe vers la page spécifique à mon panel d'administration pour les rôles dans le fichier « *sidebar.blade.php* » qui est un fichier généré par Backpack.

#### *Ajout du lien de notre vue de navigation*

```

30
31         <li><a href="{{ url('admin/role') }}"><i class="fa fa-tags"></i> <span>Roles</span></a></li>
32

```

## Résultat obtenu de notre panel d'administration sur notre site :

*Vue générale du panel d'admin des rôles*

The screenshot shows the 'Roles' section of the administration panel. The left sidebar has a 'Membres' category expanded, showing 'Users', 'Familles', 'Roles', and 'Equipes'. The main content area displays a table of roles with columns 'Titre' and 'Membres'. The table lists various roles such as 'Web Developer', 'Web Master', 'Web Communication', 'Président', 'Secrétaire', 'Manager', 'Trésorier', 'Membre CA', 'Joueur', 'Entraîneur', 'Pôle Communication', 'Pôle Financier', 'Pôle Sportif', and 'Assistant CA'. Each row includes an 'Actions' column with 'Modifier' and 'Supprimer' buttons. A search bar and a pagination link '1 Suivant' are at the bottom.

*Vue du panel d'administration lors de l'ajout d'un rôle*

The screenshot shows the 'Ajouter un rôle' (Add role) page. The left sidebar is identical to the previous screenshot. The main content area has a heading 'Nouvel ajout' and fields for 'Titre\*' (with 'Joueur' entered) and 'Membres' (with a list box). At the bottom are 'Enregistrer et retour' and 'Annuler' buttons.

*Vue du panel d'administration lors de la modification d'un rôle*

### Quelques cas particuliers :

Dans la plupart des cas, la démarche que je viens de décrire est suffisante pour que tout fonctionne à merveille. Bien entendu je suis tombé sur quelques cas particuliers qui m'ont obligé à creuser un peu plus dans la documentation de Backpack.

Un détail certes, mais qui a son importance. Quand j'étais sur le panel d'administration des utilisateurs, la liste de ces derniers m'était rendue dans l'ordre des identifiants et non par ordre alphabétique des noms. Pour modifier cela, j'ai dû me rendre dans le modèle des utilisateurs, *User.php*, et y ajouter un « scope », la fonction *boot()*, pour y modifier la manière dont les utilisateurs étaient ordonnés.

*Modification de la façon d'ordonner les utilisateurs*

```

195  protected static function boot()
196  {
197      parent::boot();
198
199      static::addGlobalScope('ordered', function(Builder $builder) {
200          $builder->orderBy('name');

```

Dans le cas où je mets des champs images (pour les utilisateurs, les équipes et les albums par exemple) l'enregistrement dans la base de données se fait de façon automatique, stockant le chemin où l'image se trouve. Par contre l'image ne se sauvegarde pas de manière automatique sur notre serveur. Pour cela, j'ai dû retourner sur mes modèles (pour cet exemple j'ai choisi *Album.php*) et y ajouter un « mutator ». Je devais respecter une certaine syntaxe pour que Backpack comprenne automatiquement quand utiliser la fonction qui permet de sauver une image sur le serveur. Pour cela je devais noter ma fonction comme ceci : « *setPhotosAttribute()* ». « set » indique à Backpack que c'est lors d'une méthode POST et « PhotosAttribute » lui indique à quel champ je fais référence. Avec cette syntaxe, backpack sait qu'il doit utiliser cette fonction plutôt que celle de base concernant le champ « photos ».

Une fois cette fonction déclarée, il ne me reste plus qu'à définir ce que je veux qu'il se passe lorsque j'appelle une méthode POST contenant le champ « photos ». Dans cette exemple je voulais récupérer chaque photo sélectionnée pour :

- lui donner des tailles spécifiques dans le but de pourvoir ajouter un « srcset » dans mes vues,
- lui donner un nom unique ainsi qu'à son album pour un stockage plus efficient,
- la stocker sur le serveur dans un fichier spécifique et par album,
- stocker le chemin public des fichiers dans la base de données sous format json.

Toutes ces étapes sont relativement simples et faisables avec des fonctions php de base alors que pour l'étape de stockage sur le serveur je suis allé modifier le fichier « *filesystem.php* » qui est généré par Laravel de base pour y insérer un disk qui me permet d'ajouter des éléments sur mon serveur.

#### Ajout du disk *filesystem.php*

```
58     'public_folder' => [
59         'driver' => 'local',
60         'root' => public_path(),
61     ],
```

*Fonction pour enregistrer les photos de l'album*

```

175     public function setPhotosAttribute($value)
176     {
177         $attribute_name = "photos";
178         $disk = "public_folder";
179         $destination_path = "uploads/albums";
180         $album_originalName = $_POST['name'];
181         foreach ( $value as $file ) {
182             // 0. Generate new size file
183             $image = \Image::make( $file );
184             $image_350 = \Image::make( $file ) -> widen( 350 );
185             // 1. Generate new file names & album name
186             $album_name = md5( $album_originalName . time() );
187             $new_file_name = md5( $file -> getClientOriginalName() . time() . '.' . $file->getClientOriginalExtension();
188             $new_file_name_350 = md5( $file -> getClientOriginalName() . time() . '_350.' . $file->getClientOriginalExtension();
189             // 2. Move new files to the correct path
190             $file_path = $destination_path . '/' . $album_name . '/' . $new_file_name;
191             $file_path_350 = $destination_path . '/' . $album_name . '/' . $new_file_name_350;
192             \Storage::disk( $disk )->put( $file_path, $image -> stream() );
193             \Storage::disk( $disk )->put( $file_path_350, $image_350 -> stream() );
194             // 3. Add the public path to the database
195             $attribute_value[] = [ $file_path, $file_path_350 ];
196         }
197         $this->attributes[$attribute_name] = json_encode($attribute_value);
198     }

```

Pour le cas de mes équipes, il peut y en avoir deux avec le même nom mais appartenant à une saison différente. Dans le cas de la création d'un match ou d'un album, je propose la possibilité de sélectionner parmi les équipes existantes. Le problème étant le suivant : seul le nom des équipes est affiché dans la liste de sélection, il m'est donc impossible de différencier deux équipes d'une saison différente ayant le même nom. Pour remédier à cela, j'ai trouvé une solution dans la documentation de Backpack. Les « accès » fonctionnent de la même façon que les « mutateurs » mentionnés plus haut avec une petite différence : au lieu de se déclencher lors qu'une méthode POST, celle-ci se déclenche lors d'une méthode GET. La syntaxe est sensiblement la même sauf que je dois remplacer le « set » par un « get », ce qui me donne « *getDivisionSeasonAttribute()* » comme nom de fonction.

*Fonction me permettant de distinguer deux équipes du même nom*

```

157     */
158     public function getDivisionSeasonAttribute( $value )
159     {
160         $division = $this->division;
161         $season = $this->season;
162         $value = $division.' / '.$season;
163         return $value;
164     }

```

Dans cette fonction, je rends disponible l'attribut « divisionSeason » qui me retournera une chaîne de caractères comprenant le nom de l'équipe et la saison à laquelle elle appartient séparé par un « / ». Une fois cette fonction définie, je n'ai plus qu'à l'appeler dans mon contrôleur pour ressortir la valeur que j'ai demandée.

*Appel de l'attribut pour un champ de création d'un album*

```

42     $this->crud->addField(
43         [
44             'label'      => 'Équipe liés à l\'album',
45             'type'       => 'select2_multiple',
46             'name'       => 'teams', // the method that defines the relationship in your Model
47             'entity'     => 'teams', // the method that defines the relationship in your Model
48             'attribute'  => 'divisionSeason', // foreign key attribute that is shown to user
49             'model'      => 'App\Models\Team', // foreign key model
50             'pivot'      => true, // on create&update, do you need to add/delete pivot table entries?
51         ],
52         'both'
53     );

```

*Rendu dans le panel de création d'un album photos*

## De l'autre côté du sac à dos

Laravel est un framework basé sur le fonctionnement de « Model-View-Controller ». C'est une méthode que j'ai rapidement vue au cours de ma formation, une façon de séparer les actions différentes dans des dossiers spécifiques.

Comme pour les panneaux d'administration, la marche à suivre est assez répétitive pour chaque page avec cependant quelques cas particuliers.

Pour intégrer une page de mon site, je dois passer par plusieurs étapes que je suis sur le point de décortiquer. Je prends comme exemple la page d'accueil puisqu'elle couvre plusieurs sortes de données en même temps.

Chaque page correspond à une url spécifique. La première chose que je fais quand je crée une nouvelle page est d'autoriser l'url de cette page en la déclarant à Laravel dans mon fichier de routes « web.php ».

*Déclaration de l'url de la page d'accueil dans le fichier de routes*

```
16     Route::get('/', 'PageController@home')->name('home');
```

Le « get » représente la méthode avec laquelle est appelée l'url. Le premier argument à déclarer est l'url, ici : « / » pour la page d'accueil. Le second argument peut être découpé en deux parties séparées par le « @ ». La première partie, « PageController », représente la classe de mon contrôleur, c'est dans cette classe que je vais répertorier des fonctions qui vont me retourner une vue. La deuxième partie de cet argument, « home », est le nom de la fonction qui va être appelée au sein de cette classe. Le dernier élément, « name('home') », est facultatif et ne sert qu'à donner un nom à cette route pour y faire référence plus facilement ultérieurement.

Une fois la route autorisée, je dois aller dans le fichier contrôleur, « *PageController.php* » que j'ai spécifié et y définir la fonction mentionnée, « home », dans le fichier de routes. Le minimum « syndical » que cette fonction doit faire c'est d'indiquer à Laravel quelle vue utiliser quand cette fonction est appelée.

*fonction « home » la plus simple possible*

```

23  public function home()
24  {
25      return view('home');
26 }
```

Une vue est un fichier HTML qui sera chargé par le navigateur quand on l'appelle. Dans le cas de Laravel, ce ne sont pas des fichiers HTML mais « Blade », la syntaxe est un peu différente mais cela reste sensiblement la même chose.

La fonction « *view('home')* » indique à Laravel d'aller chercher le fichier appelé « *home.blade.html* » dans le dossier « *views* » de l'architecture Laravel. Pour des raisons pratiques, j'ai décidé de fractionner mes fichiers de vues. Cela m'évite de répéter du code identique sur plusieurs pages et grâce à « Blade » cela me permet d'inclure des fichiers facilement.

*Fichier de vue de base nommé « layout.blade.php »*

```

1  @include( 'partials.head' )
2  @include( 'partials.nav' )
3
4  @yield( 'content' )
5
6  @include( 'partials.footer' )
7  @include( 'partials.html-footer' )
```

Nous pouvons constater que j'inclus 4 fichiers grâce à la fonction « `@include()` ». Le paramètre « `'partials.head'` » est la syntaxe pour cibler un fichier nommé « `head.blade.php` » dans un sous-dossier de « `views` » portant le nom « `partials` ». La fonction « `@yield()` » signifie que c'est à cet endroit que le contenu de la section « `content` » se placera.

Grâce à ce fichier de base, je ne dois plus spécifier que j'ajoute les 4 fichiers dans chacune de mes vues. Il me suffit d'indiquer que mon fichier est une extension du fichier de base, « `layout.blade.php` », et d'entourer le contenu de mes fichiers en spécifiant que c'est le contenu de la section « `content` » pour que Laravel sache où le placer.

*Fonction entourant mes fichiers de vues*

```
1 @extends('partials.layout')
2 @section('content')
3
4     {{-- code ici --}}
5
6 @endsection
```

Dans les cas les plus simples cette marche à suivre suffit, mais pour la plupart des cas, j'ai besoin d'utiliser des variables au sein de mes vues comme par exemple pour récupérer le titre de ma page, ou des informations provenant de ma base de données.

Pour cela, c'est dans le contrôleur que je dois définir les variables et les incorporer dans ma vue. Lorsque ce sont des variables ne provenant pas de ma base de données, il me suffit de les déclarer dans la fonction du contrôleur et de les envoyer à ma vue grâce à la fonction « `compact()` ».

*Exemple de contrôleur avec une variable transmise à la vue*

```
23 public function home(Request $request)
24 {
25     // ***** Page title *****
26     $pageTitle = "Accueil";
27
28     return view('home', compact('pageTitle'));
29 }
```

Une fois cette variable envoyée à la vue, j'ai la possibilité de l'utiliser autant que je veux dans mon fichier « Blade » avec une syntaxe particulière.

*Utilisation de la variable « pageTitle » dans la vue « home.blade.php »*

```

1  @extends('partials.layout')
2  @section('content')
3
4      <h1>{{$pageTitle}}</h1>
5
6  @endsection

```

Dans la majorité des cas, j'ai besoin d'interagir avec ma base de données pour récupérer des informations, les stocker dans des variables pour les utiliser dans ma vue. C'est là que j'ai besoin d'utiliser mes modèles pour éviter de tout faire au même endroit, toujours dans l'idée de garder une architecture claire avec des fichiers les plus simples possibles.

L'objectif est de ne rien faire d'autre que d'attribuer des valeurs à des variables et les envoyer à ma vue quand je suis dans mon fichier contrôleur. Pour cela, je fais en sorte d'appeler des fonctions qui sont définies dans mes modèles. J'ai donc besoin d'aller dans le modèle « Sponsor » et y définir la fonction dont j'ai besoin, « `getAllSponsors()` ».

*Définition de la fonction dans mon modèle « Sponsor.php »*

```

34  public function getAllSponsors()
35  {
36      $sponsors = DB::table('sponsors') -> get();
37
38      foreach( $sponsors as $sponsor ){
39
40          // ***** Get the image srcset *****
41
42          $img = $sponsor->image;
43
44          $imgSplited = preg_split( '/\./', $img );
45
46          $imgName = $imgSplited[0];
47
48          $imgExt = $imgSplited[1];
49
50          $sponsor->srcset = $imgName.'._200.'.$imgExt.' 200w';
51
52      }
53
54
55      return $sponsors;
56  }

```

Cette fonction me retourne un objet contenant tous les sponsors encodés dans la base de données ainsi que les informations qui leur sont propres.

Une fois la fonction définie, je demande à mon contrôleur de récupérer les valeurs qu'elle me retourne. Dans cet exemple, je crée une instance de mon modèle « Sponsor » dans mon contrôleur pour avoir accès aux fonctions définies dans ce modèle. Une fois cette instance créée, j'appelle la fonction dont j'ai besoin et je stocke le résultat dans une variable que j'envoie à ma vue grâce à la fonction « *compact()* » dont j'ai parlé plus haut.

#### *Exemple de variables définies à partir d'une instance*

```

49     // ***** Get all sponsors *****
50     $sponsors = new Sponsor;
51     $sponsors = $sponsors -> getAllSponsors();

```

Voici donc comment ce système de « Model-View-Controller » fonctionne dans la plupart des cas de mon projet. Je sais dorénavant indiquer à Laravel quel fichier le navigateur doit charger en fonction de quelle url il reçoit. Je sais comment faire pour passer des variables de mon contrôleur à ma vue, mais il y a des variables qui sont constantes à toutes les vues et les redéfinir à chaque fois serait une grande perte de temps et de performance. J'ai donc été chercher dans la documentation de Laravel pour trouver un moyen de passer une variable à toutes les vues à la fois.

Je dois agir dans le fichier « *AppServiceProvider.php* » généré par Laravel. Il me suffit de définir des variables au sein de la fonction « *boot()* » et de la partager dans chaque vue. Dans ce cas, j'affiche l'adresse e-mail du club dans le pied de page. Pour ne pas l'inscrire en dur, c'est une donnée que j'ai stockée dans ma table « *settings* » dans ma base de données. Le pied de page se trouvant sur chaque page, j'avais besoin que ma variable contenant l'adresse e-mail du club soit accessible sur chacune de mes vues.

#### *Définition de la variable \$contactEmail dans la fonction « boot() »*

```

36     // ***** Get the contact email on every view *****
37     $DB_contactEmail = DB::table('settings') -> select('value') -> where('key', 'contact_email') -> get();
38     $contactEmail = $DB_contactEmail[0]->value;
39     View::share('contactEmail', $contactEmail);

```

Dans la fonction `boot()` du fichier « `AppServiceProvider.php` », je récupère l'information de la base de données dont j'ai besoin pour la stocker dans une variable. Une fois cette variable définie, il ne me reste plus qu'à la partager à toutes mes vues grâce à la fonction « `share()` » de la classe « `View` » de Laravel.

C'est avec cette méthodologie que j'ai codé toutes mes vues, contrôleurs et modèles. C'est un système assez répétitif et rapidement pris en mains avec une architecture bien pensée dans laquelle je n'ai pas eu trop de mal à me retrouver.

## Des navigateurs sans Javascript, ça existe encore ?

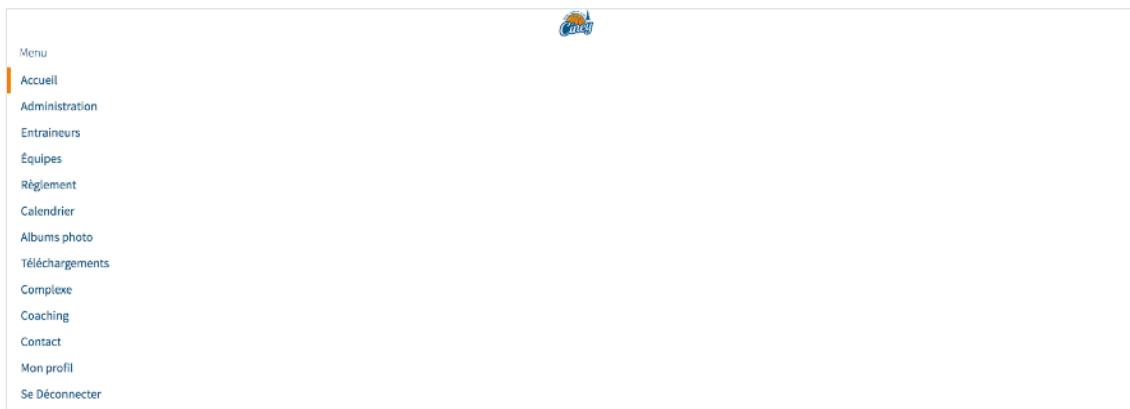
Un des critères m'ayant été imposé pour ce travail était qu'il soit entièrement fonctionnel sans Javascript. Beaucoup diraient que ce n'est pas nécessaire car tous les navigateurs sont dotés de Javascript de nos jours et qu'il n'y a donc aucun intérêt à « s'embêter avec ce genre de fioriture », et c'est ce que je pensais aussi au début. Avec un peu de recul je trouve cela intéressant d'avoir prévu un quelconque problème de script. C'est en effet tout à mon avantage et c'est un atout de plus pour mon projet. Dans cette idée-là, la marche à suivre est simple, j'ai intégré mon design sans aucun javascript dans un premier temps puis je suis repassé dessus pour en ajouter.

L'objectif est simple : se rapprocher le plus possible du design en utilisant uniquement du CSS. Il va de soi que c'est une tâche impossible car quelques éléments nécessitent l'utilisation de javascript pour avoir le rendu désiré mais il faut faire en sorte de s'en rapprocher le plus possible.

### Quelques exemples d'éléments adaptés à une utilisation sans javascript :

Dans le cas d'un menu déroulant, pour qu'il soit accessible j'avais besoin d'utiliser du javascript, j'ai donc décidé d'afficher le menu en entier et sans avoir de menu déroulant lorsque le javascript ne fonctionnait pas.

*Apparence du menu sans javascript*



Ceci est moins « beau » et prend beaucoup plus de place mais c'est surtout entièrement fonctionnel même quand les scriptes ne se chargent pas.

Pour ce qui est des sliders d'images que j'avais prévus dans mon design, il m'était impossible de les implémenter sans javascript, j'ai donc décidé de les disposer de la même façon dans leur container avec des propriétés CSS permettant un scroll horizontal pour donner un semblant de slider.

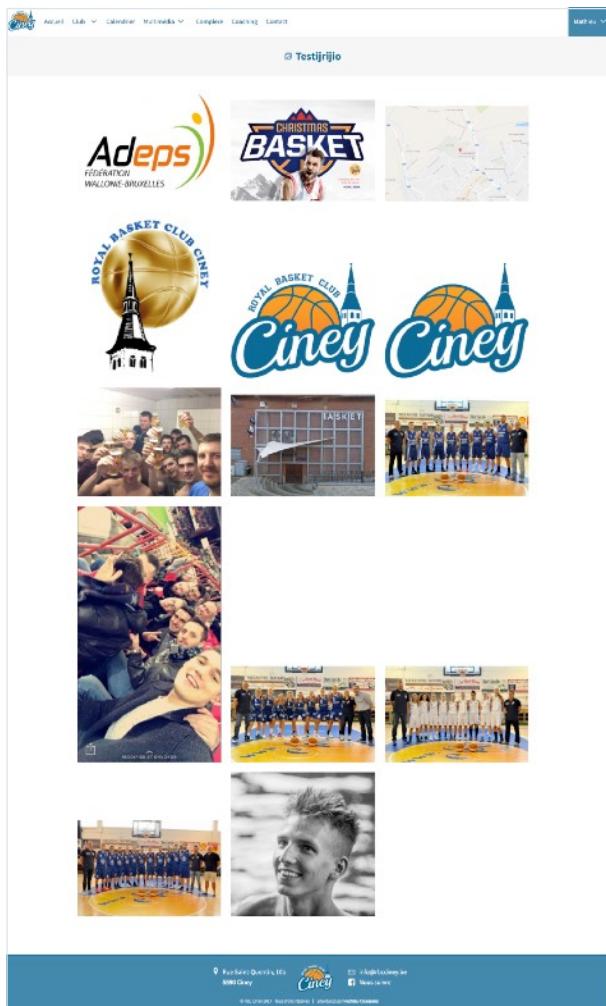
*Affichage d'un slider sans javascript*



Dans le cas des paginations pour les matchs et résultats, elles sont fonctionnelles mais entraînent un chargement complet de la page, ce qui n'est pas optimal en terme de performance mais encore une fois, cela fonctionne sans scripte, c'est le plus important.

Concernant les murs de photos sur la page dédiée à un album, j'ai utilisé un « display: flex; » pour avoir un agencement des photos le plus propre possible. Lorsqu'il y a des photos de dimensions très variées, il y a des « trous » dans ce mur de photos mais le résultat reste acceptable.

*Affichage du mur de photos sans javascript*



Dans la page de règlements, on trouve une navigation interne. J'ai choisi de l'afficher par défaut en « position: absolute; » plutôt qu'en « position: sticky; » pour éviter tout problème dans le cas où la navigation serait plus longue que l'écran de l'utilisateur et par conséquent, serait en partie cachée et inutilisable. Grâce à ce positionnement, l'utilisateur peut librement scrolller la page pour afficher le reste de la navigation. Le point négatif de cette façon de fonctionner, c'est que la navigation reste sur le haut de la page et n'est donc plus visible une fois que l'utilisateur a scrollé tout en bas.

La navigation n'est pas non plus interactive, il n'y a aucun indicateur dans la navigation de la position dans le contenu de la page. C'est un détail qui rend l'expérience utilisateur plus agréable, mais encore une fois, impossible sans javascript.

*Aperçu de la navigation interne aux règlements en haut de la page*

The screenshot shows a website layout with a sidebar on the right containing a vertical list of navigation links. The main content area has several sections: "Règlement d'ordre intérieur", "Introduction", "Entraînements", "Matches", and "Navigation".

- Règlement d'ordre intérieur**
- Introduction**
- Entraînements**
  - La participation aux entraînements est indispensable.**
  - Le joueur qui pour une raison sérieuse [cas de force majeure, maladie, étude, blessure, travail, ...] n'est pas en mesure de participer à un entraînement devra en informer son entraîneur **au plus tard 2 heures** avant le début de celui-ci. Il est vivement conseillé d'arriver **1/4 d'heure avant le début de l'entraînement** afin de se préparer calmement et méthodiquement.
  - L'entraînement débutera **exactement à l'heure prévue**.
  - Pour profiter au maximum du temps qui vous est imparti et pour que l'entraînement puisse se dérouler dans les meilleures conditions :
    - une concentration totale sera exigée.
    - il y aura lieu de s'arrêter immédiatement aux injonctions de l'entraîneur.
- Matches**
- Navigation**
  - Règlement d'ordre intérieur
  - Règlement de la salle J Laloux (parquet)
  - Règlement interne du BC
  - Avis à tous les membres du club
  - Cotisation
  - Déclaration d'accident
  - Cerificat médical
  - Inscription dans une équipe
  - Etre parent au RBC Ciney

*Aperçu de la navigation interne aux règlements après avoir scrollé*

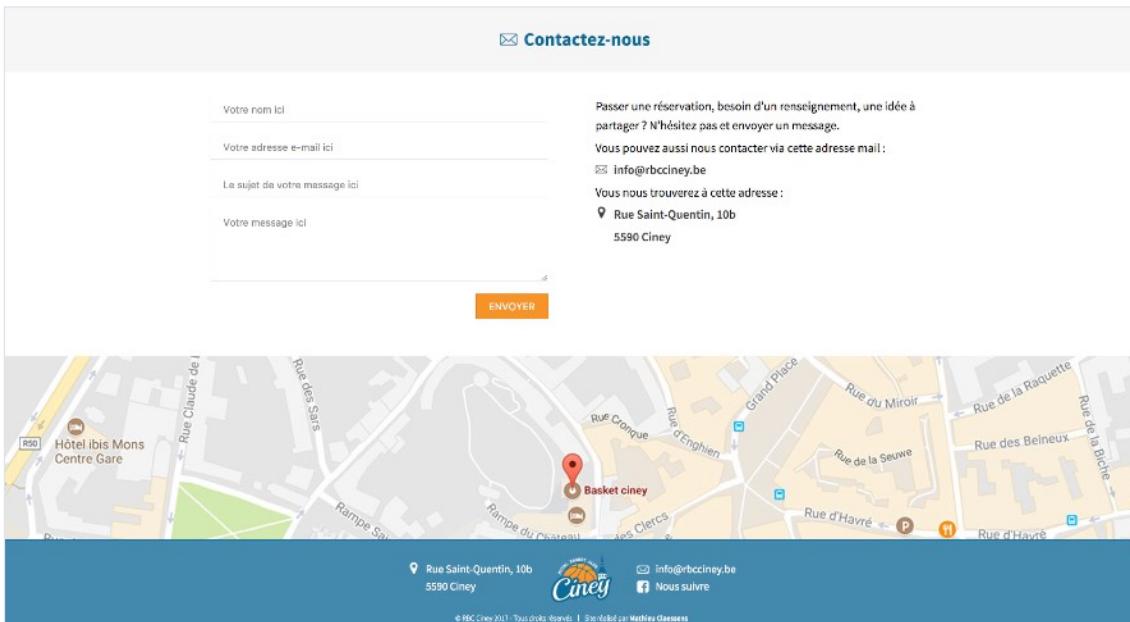
The screenshot shows the same website layout after scrolling down. The sidebar on the right still contains the "Navigation" link, but the main content area now displays the "Entraînements" section in full, while the other sections ("Matches", etc.) are partially visible at the top of their respective columns.

<b>Entraînements</b>	Avis à tous les membres du club
<b>La participation aux entraînements est indispensable.</b>	Cotisation
Le joueur qui pour une raison sérieuse [cas de force majeure, maladie, étude, blessure, travail, ...] n'est pas en mesure de participer à un entraînement devra en informer son entraîneur <b>au plus tard 2 heures</b> avant le début de celui-ci. Il est vivement conseillé d'arriver <b>1/4 d'heure avant le début de l'entraînement</b> afin de se préparer calmement et méthodiquement.	Déclaration d'accident
L'entraînement débutera <b>exactement à l'heure prévue</b> .	Cerificat médical
Pour profiter au maximum du temps qui vous est imparti et pour que l'entraînement puisse se dérouler dans les meilleures conditions :	Inscription dans une équipe
<ul style="list-style-type: none"> <li>• une concentration totale sera exigée.</li> <li>• il y aura lieu de s'arrêter immédiatement aux injonctions de l'entraîneur.</li> </ul>	Etre parent au RBC Ciney
<b>Matches</b>	
<b>La participation à tous les matchs est indispensable.</b>	
Le joueur qui pour une raison sérieuse, n'est pas en mesure de participer à un match devra en avertir son coach le plus rapidement possible.	
Sachez que votre sélection repose sur divers facteurs dont notamment:	
<ul style="list-style-type: none"> <li>• la régularité aux entraînements.</li> <li>• la ponctualité.</li> <li>• le travail fourni aux entraînements [la volonté de progresser].</li> <li>• la forme du moment.</li> <li>• la tactique employée.</li> <li>• la discipline.</li> </ul>	

Dans mon design j'avais prévu d'intégrer une « google map » interactive sur la page de contact ainsi que sur la page du complexe sportif. Malheureusement l'intégration d'une « google map » sur mon site requiert l'utilisation de javascript. Il m'est donc impossible d'en intégrer une dans un premier temps.

Je souhaitais mettre quand même un élément dans ma page qui simulerait une « google map » dans le cas où le javascript ne se chargerait pas. J'ai donc décidé d'ajouter une image prise à partir de « google map » centré sur la position du club et de l'intégrer à la place de la « google map ». Malheureusement une image n'est pas interactive, j'ai décidé de la rendre cliquable pour rediriger l'utilisateur vers le site de google map avec la position du club déjà encodée.

## *Aperçu de la google map sans javascript*



## Il est venu le temps du Javascript

Une fois tout mon site codé et fonctionnel sans javascript, il est temps de s'amuser un petit peu et de revoir la vie du bon côté, du côté javascript.

Devoir être fonctionnel sans javascript ne veut pas dire ne pas utiliser de javascript du tout, au contraire. Un site qui fonctionne sans javascript est une bonne chose mais quand il y a la possibilité de l'utiliser pourquoi s'en priver ?

Le tout premier élément sur lequel j'ai ajouté du javascript a été mon menu. Mr Vilain m'avait parlé du site d'adobe qui développait un système de menu respectant les règles d'accessibilité du web en terme de menu navigable sans souris. J'ai donc été voir ce qu'il faisait et ce qu'il proposait. Il s'agissait d'un plugin « accessibleMegaMenu » qui générerait un menu accessible avec « dropdown » si je respectais la mise en place de ce dernier.

Un plugin très intéressant mais qui pour moi ne fonctionnait pas pour le design mobile. J'ai donc décidé de coder un menu burger par moi-même pour la version mobile puis d'utiliser le plugin pour la version desktop.

J'ai commencé par appliquer des classes différentes à ma navigation en fonction de la taille d'écran pour faciliter le transfert entre les deux affichages.

*Code qui applique la classe en fonction de la taille de l'écran*

```
30 // ***** Nav responsive *****
31 function responsiveNav(width){
32     $('.nav:first').removeClass('desktop').removeClass('mobile');
33     $('.burgerButton').removeClass('desktop').removeClass('mobile');
34     $('.nav-menu').removeClass('nav-menu-desktop');
35     if (width >= 980) {
36         $('.nav:first').addClass('desktop');
37         $('.nav-menu').addClass('nav-menu-desktop');
38     } else {
39         $('.nav:first').addClass('mobile');
40         $('.burgerButton').addClass('mobile');
41     }
42 } responsiveNav(windowWidth);
```

Cela établi, j'ai fait en sorte que mon menu burger fonctionne comme il se doit pour la version mobile. Lorsque l'utilisateur clique sur le bouton de menu, cela déclenche une série d'événements. Cela ajoute une classe « nav-open » au <body> de ma page ainsi qu'au bouton cliqué et le <nav> qui le suit. Si la classe est déjà présente, elle est supprimée, cela fonctionne comme un interrupteur. Ensuite je fais en sorte de retirer toutes les classes « nav-open » lorsqu'on clique en dehors de ma navigation.

*Code exécuté lors du clic sur le bouton*

```
18 // ***** Burger button *****
19 $('.burgerButton').click(function(e) {
20     e.preventDefault();
21     $(this).toggleClass('nav-open');
22     $('body').toggleClass('nav-open');
23     $(this).next().toggleClass('nav-open');
24     $('body').click( function() {
25         $('.nav-open').removeClass('nav-open');
26     });
27     e.stopPropagation();
28 });

});
```

C'est tout ce qu'il faut pour que mon menu burger fonctionne sachant que j'ai prévu un style CSS spécifique lorsque la classe « nav-open » est présente.

*Rendu du menu burger*

The screenshot shows a contact form on a website. At the top left is the club's logo. In the center, there's a title "✉ Contactez-nous". Below it, there's text about contacting via email or message. Further down, there are fields for name, email, subject, and message. On the right side, there's a sidebar with a list of links: Accueil, Administration, Entraineurs, Équipes, Règlement, Calendrier, Albums photo, Téléchargements, Complexe, Coaching, Contact (highlighted in orange), Mon profil, and Se Déconnecter.

Concernant le menu desktop, dont la classe sera ajoutée grâce à mon scripte plus haut, je déclenche le plugin de « accessibleMegaMenu ». La documentation étant assez claire, je n'avais qu'à préciser quelques paramètres à la fonction; le tour était joué. J'ai spécifié la classe de ma navigation globale dans l'argument « menuClass » en y indiquant la classe que j'ajoute en javascript lorsque l'écran a une taille desktop pour éviter de déclencher le plugin quand je suis en format mobile. Il m'a suffi ensuite de déclarer les classes que j'avais choisies pour les différents éléments de ma navigation telles que la classe de mes éléments de navigation principale, « nav-item », ainsi que celles des sous-navigations, « sub-nav », dans le cas de menu « dropdown » et leur navigation, « sub-nav-group ». J'ai laissé les autres paramètres par défaut car ils ne dépendaient pas de mon code.

#### Structure de mon fichier de navigation

```

14   <nav>
15
16     <ul class="nav-menu">
17
18       <li class="nav-item{{ Request::is('/') ? ' active' : '' }}>
19
20         <a class="" href="/" title="Retour à l'accueil">Accueil</a>
21
22       </li>
23
24       <li class="nav-item">
25
26         <a class="dropdown" href="#" title="Voir les liens de la rubrique club">Club</a>
27
28         <div class="sub-nav">
29
30           <ul class="sub-nav-group">
31
32             <li class="sub-nav-item{{ Request::is('conseil-administration') ? ' active' : '' }}>
33
34               <a class="icon icon-admin" href="/conseil-administration" title="Voir notre conseil d'administration">Administration</a>
35
36             </li>
37
38             <li class="sub-nav-item{{ Request::is('entraineurs') ? ' active' : '' }}>
39
40               <a class="icon icon-trainer" href="/entraineurs" title="Voir nos entraîneurs">Entraîneurs</a>
41
42             </li>
43
44             <li class="sub-nav-item{{ Request::is('equipes') ? ' active' : '' }}>
45
46               <a class="icon icon-team" href="/equipes" title="Voir nos équipes">Équipes</a>
47
48             </li>
49
50             <li class="sub-nav-item{{ Request::is('regles') ? ' active' : '' }}>
51
52               <a class="icon icon-rule" href="/regles" title="Voir notre règlement">Règlement</a>
53
54             </li>
55
56           </ul>
57
58         </div>
59
60       </li>
61
62       <li class="nav-item{{ Request::is('calendrier') ? ' active' : '' }}>
63
64         <a class="" href="/calendrier" title="Voir notre calendrier">Calendrier</a>
65
66       </li>

```

### *Initialisation du plugin*

```

44 // ***** Initialize accessibleMegaMenu *****
45 $('nav:first').accessibleMegaMenu({
46   // prefix for generated unique id attributes, which are required to indicate aria-owns, aria-controls and aria-labelledby
47   uidPrefix: 'accessible-megamenu',
48   // css class used to define the megamenu styling
49   menuClass: 'nav-menu-desktop',
50   // css class for a top-level navigation item in the megamenu
51   topNavItemClass: 'nav-item',
52   // css class for a megamenu panel
53   panelClass: 'sub-nav',
54   // css class for a group of items within a megamenu panel
55   panelGroupClass: 'sub-nav-group',
56   // css class for the hover state
57   hoverClass: 'hover',
58   // css class for the focus state
59   focusClass: 'focus',
60   // css class for the open state
61   openClass: 'open'
62 });

```

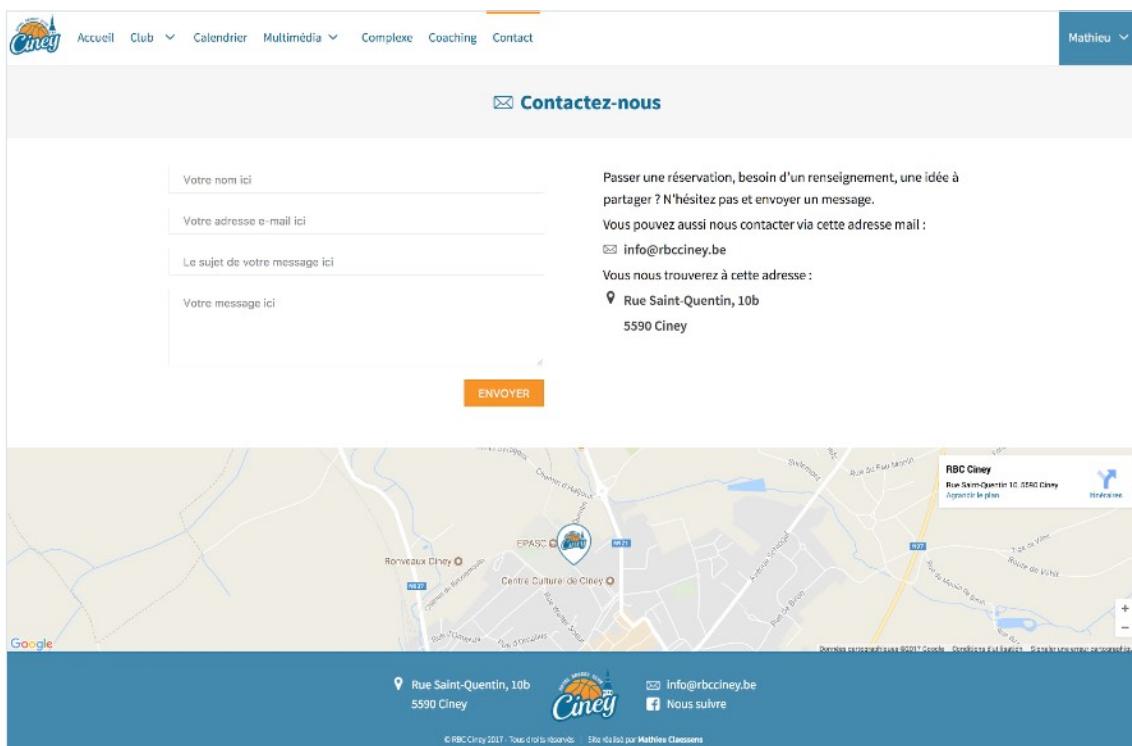
J'avais un menu desktop parfaitement accessible, navigable sans aucune souris, avec seulement un clavier ou encore avec un programme de lecture. Il ne me restait plus qu'à y ajouter ma touche personnelle en CSS pour qu'il corresponde à mon design.

### *Rendu du menu une fois le plugin initié*



Le deuxième élément dont j'ai déjà parlé plus haut et pour lequel j'ai dû utiliser du javascript est la « google map ». J'ai donc suivi la documentation de google directement, un setup relativement classique mis à part que j'ai personnalisé le marqueur de la carte avec un icon que j'ai créé personnellement et j'ai ajouté la petite carte avec l'adresse du club, le lien vers l'itinéraire et un autre pour agrandir la carte. J'ai bien entendu désactivé le scroll en ajoutant le paramètre « scrollwheel : false » pour éviter que les utilisateurs ne puissent atteindre la fin de ma page parce qu'ils sont en train de scroller dans la « google map ».

### *Rendu de la « google map » sur la page contact*



Concernant les albums photos, sans javascript, je ne pouvais pas m'assurer que mon mur de photos soit homogène lorsque l'album contient des photos de formats trop différents, cela crée des « trous ». Pour remédier à cela, j'ai utilisé le plugin « masonry ». Il permet de changer l'agencement des photos pour former un mur le plus homogène possible. Ce plugin est bien entendu responsive sans quoi il ne serait pas intéressant. Je n'avais qu'à retirer les styles que j'avais appliqués en retirant les classes initiales pour ne laisser que celles dédiées au plugin « masonry ». Je n'avais qu'à spécifier les classes cibles dans les options ainsi que le positionnement en pourcentage, un choix personnel pour rester dans la logique de mon code.

*Code retirant les classes et lançant le plugin « masonry »*

```
102 // ***** Masonry *****
103 if ($('body').hasClass('page-albums')) {
104     // Remove default class if JS enable
105     $('.album-wrap').removeClass('album-wrap');
106     $('.album-photo').removeClass('album-photo');
107     // Initiate Masonry
108     $('.masonry').masonry({
109         // Options
110         itemSelector: '.masonry-item',
111         percentPosition: true
112     });
113 }
```

*Rendu du mur de photos grâce au plugin « masonry »*



Passons maintenant aux sliders. J'avais fait en sorte de disposer d'un scroll horizontal pour avoir un semblant de slider dans le cas où il n'y avait pas de javascript. Il fallait que je maquille tout ça en un beau slider interactif et responsive. Pour m'aider à réaliser cette tâche, j'ai choisi d'utiliser un plugin, « Slick », dont je m'étais déjà servi lors de mon stage chez Exit Studio et qui m'avait été recommandé par Julien pour un de leurs projets. Ce plugin offre pas mal d'options diverses pour personnaliser les sliders au maximum. Encore une fois, j'ai fait en sorte de retirer les classes de base avant d'appliquer celle du plugin. Pour le slider des sponsors présents sur la page d'accueil, par exemple, je voulais que ceux-ci défilent automatiquement, il m'a suffi de spécifier le paramètre « autoplay: true » dans la fonction de « slick ».

#### *Aperçu du code pour le slider des sponsors sur la page d'accueil*

```

117     // Remove default class if JS enable
118
119     $('.sponsor-wrap').removeClass('flex-wrap');
120
121     $('.sponsor-wrap').removeClass('sponsor-wrap');
122
123     $('.sponsor').removeClass('sponsor');
124
125     $('.sponsor-img').removeClass('sponsor-img');
126
127     $('.complexe-album-flex').removeClass('complexe-album-flex');
128
129     // Initiate Slick for sponsors
130
131     $('.sponsor-slider').slick({
132         dots: false,
133         centerMode: true,
134         slidesToShow: 3,
135         slidesToScroll: 1,
136         autoplay: true,
137         autoplaySpeed: 3000,
138         responsive: [
139             {
140                 breakpoint: 640,
141                 settings: {
142                     slidesToShow: 2
143                 }
144             },
145             {
146                 breakpoint: 480,
147                 settings: {
148                     slidesToShow: 1
149                 }
150             }
151         ]
152     });

```

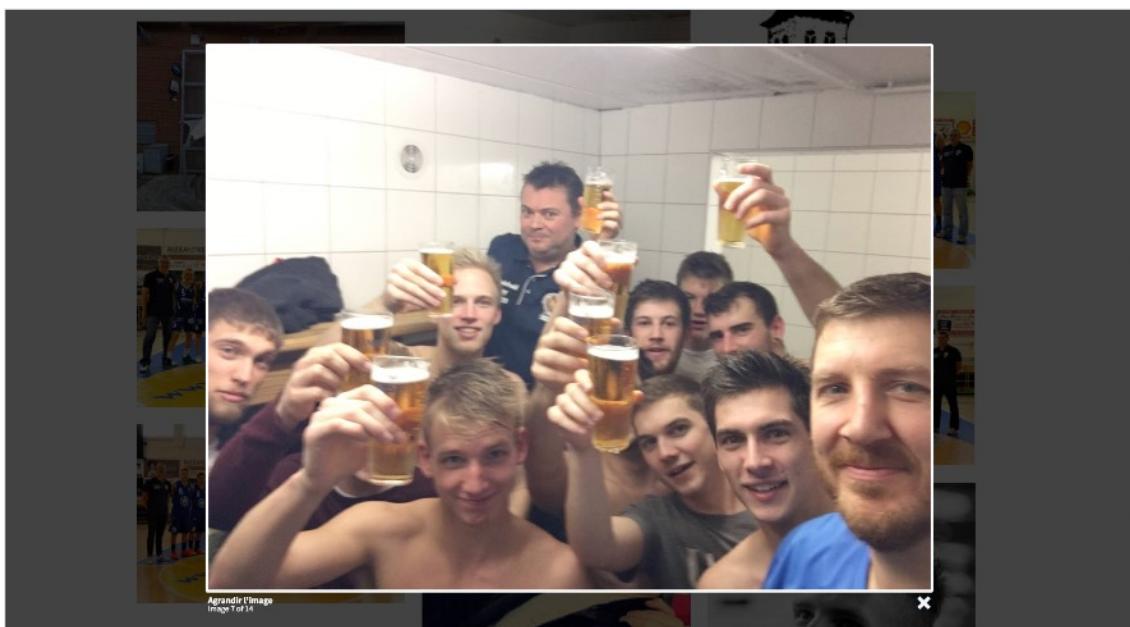
J'ai choisi le mode de slider centré puisqu'il correspondait mieux à mon design. J'ai enlevé les points de navigation qui sont affichés par défaut. J'ai dû préciser le paramètre « responsive » pour que le slider réponde favorablement au changement de taille. Dans cet exemple, plus l'écran devient petit, moins le nombre de sponsors affichés en même temps est grand jusqu'à arriver à un sponsor à la fois.

*Rendu du slider des sponsors*



Une fois les sliders mis en place, j'ai ajouté la possibilité de pouvoir cliquer sur chaque image pour la voir en taille réelle. Pour cela, j'ai utilisé un plugin des plus connus, « lightbox ». L'utilisation est particulièrement simple, il m'a suffi d'ajouter un attribut « data-lightbox » aux éléments que je voulais afficher en taille réelle. La valeur de cet attribut servait à regrouper ou non un ensemble de photos pour permettre de passer de l'une à l'autre sans sortir de l'aperçu en plein écran.

*Rendu de l'affichage en taille réelle*



Le plus gros morceau de mon script est dédié à la navigation interne de la page des règlements. Comme expliqué plus haut, par défaut, son positionnement est absolu et ne suit donc pas le contenu lors d'un scroll sur la page. Pour une raison d'accessibilité, je souhaitais que cette navigation reste à l'écran tout le long du scroll jusqu'à arriver à la fin du contenu. Dans ce but, il me fallait détecter la position actuelle de la navigation interne par rapport au scroll de la page, en fonction de la position du scroll par rapport à la navigation, je lui ajoute une classe qui lui applique un certain style.

#### *Fonction de positionnement de la navigation interne*

```

159     function currentNavbarPosition(el){
160
161         var y = $(window).scrollTop(), // Position of the scroll
162             avp = el.parent().offset().top, // Vertical position of the element container
163             ah = el.parent().height(), // Height of element container
164             navh = el.height(); // Height of navbar container
165
166         if(y >= avp - 20 ) {
167
168             // if scroll position reach the end of the container with the element height as offset
169             if(y > (avp + ah - navh - elTop - 20 )) {
170
171                 el.removeClass('fix-aside');
172
173                 el.addClass('fix-aside-bottom');
174
175                 el.css('top', (ah - navh + 45) +'px');
176
177             } else {
178
179                 el.addClass('fix-aside');
180
181                 el.removeClass('fix-aside-bottom');
182
183                 el.css('top','');
184
185             }
186
187         } else {
188
189             el.removeClass('fix-aside');
190
191         };
192
193     }

```

Une fois définie, j'ai fait en sorte que cette fonction soit appelée une première fois, je l'ai ensuite appelée à chaque fois que la page était redimensionnée car les dimensions changent et donc un nouveau calcul des valeurs est nécessaire. J'ai ensuite appelé la fonction à chaque fois qu'on scrolle dans la page pour vérifier sa nouvelle position.

*Appel de la fonction dans les différents cas*

```

180     function navbarScroll(el) {
181
182         el = $(el);
183
184         elTop = el.position().top;
185
186         currentNavbarPosition(el);
187
188         $(window).resize(function () {
189             currentNavbarPosition(el);
190         });
191     }

```

Cela étant défini, ma navigation interne suivait le contenu de la page au scroll et s'arrêtait quand elle arrivait à la fin de son container. Il me restait encore à gérer le cas où la navigation était plus longue que la hauteur du navigateur.

*Fonction appliquant une hauteur maximum à ma navigation interne*

```

195     var $root = $('html, body'),
196
197         pageHeight = $( window ).innerHeight(),
198
199         maxHeightValue = pageHeight - 160,
200
201         maxHeightString = 'max-height: ' + maxHeightValue + 'px';
202
203
204         $('.inner-nav-list').attr( 'style', maxHeightString );
205
206
207         $( window ).resize( function(){
208
209             pageHeight = $( window ).innerHeight(),
210
211             maxHeightValue = pageHeight - 160,
212
213             maxHeightString = 'max-height: ' + maxHeightValue + 'px';
214
215             $('.inner-nav-list').attr( 'style', maxHeightString );
216         } );

```

Pour assurer une meilleure expérience utilisateur, j'ai décidé d'ajouter trois fonctionnalités supplémentaires à ma navigation interne. La première étant de différencier le lien que l'utilisateur vient de cliquer par rapport aux autres. J'ai donc codé une simple fonction qui ajoute une classe à l'élément cliqué et retire cette classe à tous les autres éléments la possédant déjà.

*Fonction de différenciation d'un élément lors d'un clic*

```
245     //Add active class on click
246
247     $('.inner-nav-link').click( function(){
248
249         $('.inner-nav-link').removeClass( 'inner-nav-link-active' )
250
251         $( this ).addClass( 'inner-nav-link-active' );
252
253     });
254 }
```

La deuxième fonctionnalité dédiée à une meilleure expérience utilisateur est de rendre le mouvement de la page plus fluide lorsque la page scrolle automatiquement vers la section sélectionnée par l'utilisateur. Cela lui permet de garder ses repères lors du mouvement et d'éviter d'être perdu après avoir cliqué sur un lien dans la navigation.

*Fonction fluidifiant le scroll*

```
251     // Smooth scroll
252
253     $('.inner-nav-link').click(function() {
254
255         var href = $(this).attr('href');
256
257         $root.stop().animate({ scrollTop: $(href).offset().top
258
259             }, 500, function () {
260
261                 window.location.hash = href;
262
263             });
264
265             return false;
266
267         });
268
269     });
270 }
```

La troisième fonctionnalité m'a demandé plus de travail et de réflexion. Je souhaitais que la navigation indique l'endroit où l'utilisateur se trouvait dans la page lorsqu'il faisait défiler le contenu sans utiliser la navigation. Cela donne un feedback à l'utilisateur lors de sa navigation et facilite la compréhension de la page.

Pour arriver à mes fins, je devais donc trouver un moyen de détecter quand un élément de mon contenu était affiché sur la page pour appliquer une classe « active » à l'élément correspondant dans la navigation interne. Après de longues recherches j'ai décidé d'utiliser le plugin « Waypoint ». Ce plugin me permettait de détecter quand un élément spécifique passait à une certaine hauteur de scroll, je pouvais aussi spécifier si je voulais le détecter quand l'utilisateur scrollait vers le bas ou vers le haut.

Dans un premier temps, je devais récupérer les « id » de chaque section répertoriée par la navigation et les stocker dans un tableau « aSectionID ». Cela étant fait, je parcourait mon tableau reprenant chaque section et j'appelais une instance du plugin « Waypoint » pour chacune d'entre elles. Grâce au paramètre « element », je précisais au plugin quel élément il devait cibler. Le paramètre « handler » me permettait ensuite de définir le comportement déclenché par le passage de cet élément à un point précis du scroll. Le dernier paramètre « offset » précise à quel niveau du scroll la fonction définie dans le « handler » se déclenche, c'est pour un meilleur rendu que j'ai appliqué deux « offset » différents en fonction de la direction du scroll.

#### *Script adaptant les éléments de la navigation en fonction du scroll*

```

209     // Update class active on scroll
210
211     var aSectionID = [];
212
213     // get sections ID
214
215     $('.inner-nav-link').each( function(){
216
217         var iSectionID = $( this ).attr( 'href' ).replace( '#', '' );
218
219         aSectionID.push( iSectionID );
220
221     } );
222
223     for (var i = 0; i < aSectionID.length; i++) {
224
225         var iOffset = 0;
226
227         iOffsetDown = 100,
228
229         iOffsetUp = $( '#' + aSectionID[i] ).outerHeight() + 0.5;
230
231         // Update class active on the way down
232
233         waypoint = new Waypoint({
234
235             element: document.getElementById( aSectionID[i] ),
236
237             handler: function( direction ) {
238
239                 var test = '.link-' + this.element.id;
240
241                 $('.inner-nav-link-active').removeClass( 'inner-nav-link-active' );
242
243                 $( test ).addClass( 'inner-nav-link-active' );
244
245                 if (direction == 'down') {
246
247                     iOffset = iOffsetDown;
248
249                 } else if (direction == 'up') {
250
251                     iOffset = iOffsetUp;
252
253                 }
254
255             },
256
257             offset: iOffset
258         });
259     }
260

```

## Ajax, une formule sans recharge

Une fois la navigation gérée, j'avais fait tout ce que je souhaitais en javascript. Je suis donc passé à un autre aspect de mon site: les requêtes en Ajax. Sur ma page de calendrier, par exemple, dans la section des prochains matchs, je propose d'afficher les futures rencontres. Laravel propose un système de pagination directement lors de la récupération des données, une facilité dont j'ai profité.

### *Setup de la pagination avec Laravel*

40

```
$games = DB::table('games') -> where('date', '>=', $date) -> where('team_id', '=', $id) -> orderby('date') -> simplePaginate(6, ['*'], 'games');
```

Dans mon design je souhaitais n'avoir que les boutons « suivant » et « précédent » sans les liens vers chaque page spécifique. C'est pourquoi j'utilise la fonction « *simplePaginate()* » qui me permet de décider combien d'éléments je veux par page, quelles colonnes je vais récupérer de la base de données et de donner un nom spécifique aux paginations. Dans cet exemple, j'ai décidé d'avoir 6 matchs par page, de récupérer toutes les colonnes de ma base de données et d'appeler mes pages « games » pour les différencier de la pagination des résultats. Dans l'état actuel de mon code, Laravel a prévu une pagination pour mes matchs, lorsque je clique sur le lien « suivant », la page recharge entièrement en affichant les matchs suivants. Le problème est que la page entière est rechargée par le navigateur alors que seulement les 6 éléments de la section « prochains matchs » ont changé, cela n'est pas efficace en terme de performance, je souhaitais donc ne recharger que la section « prochains matchs » sans que le reste de la page ne bouge. Pour cela je devais faire une requête Ajax.

Je déclenche l'événement lorsque je clique sur le lien vers la page suivante ou précédente, j'annule le comportement par défaut du lien et dans un premier temps, je cache le contenu de ma section et je la remplace par un icône de chargement.

Cela étant fait, je récupère l'url de mon lien, j'appelle la fonction Ajax qui me permet de récupérer le contenu suivant et de l'afficher sans charger ma page. Je termine par changer la valeur de l'url pour qu'elle corresponde tout de même au contenu se trouvant sur la page.

Dans ma fonction Ajax, je lui indique l'url que j'ai récupérée de mon lien et je lui demande de m'afficher la réponse de la requête dans ma section « prochains matchs » grâce à la fonction « `$('.gameCard-wrap').html(data)` ». Les autres fonctions ne me servent qu'à adapter le style de ma page en fonction du contenu.

### *Code de ma fonction Ajax*

```

7  ${function() {
8      $('body').on('click', '.gameCard-wrap .pagination a', function(e) {
9          e.preventDefault();
10         $('.gameCard').css('opacity', '0');
11         $('.gameCard-wrap').append('<div class="pagination-loading"></div>');
12
13         var url = $(this).attr('href');
14         getGames(url);
15         window.history.pushState("", "", url);
16     });
17
18     function getGames(url) {
19         $.ajax({
20             url : url
21         }).done(function (data) {
22             $('.gameCard-wrap').html(data);
23             $('.gameCard-wrap a[rel="prev"]').html('Match précédent').addClass('button button-more button-more-prev');
24             $('.gameCard-wrap a[rel="next"]').html('Match suivant').addClass('button button-more button-more-next');
25             if($('.gameCard-wrap a[rel="prev"]').hasClass('button-more-prev')) {
26                 $('.gameCard-wrap').css('padding-top', '10rem');
27             } else {
28                 $('.gameCard-wrap').css('padding-top', '4rem');
29             }
30         }).fail(function () {
31             alert('Impossible de charger les matchs.');
32         });
33     }
34 });

```

Après avoir mis en place ma fonction Ajax, il ne me restait plus qu'à indiquer la valeur de la réponse de cette fonction. Pour cela, je devais me rendre dans le contrôleur de ma page de calendrier, « `PageController.php` », dans la fonction dédiée à cette page, « `calendar()` ». Je souhaitais que la réponse de ma fonction Ajax soit du code HTML affichant les matchs de ma page spécifiée. Pour cela, je devais renvoyer une vue avec les matchs à afficher et utiliser la fonction « `render()` » pour renvoyer le code HTML directement. Il me fallait donc détecter le cas où une requête Ajax était envoyée et retourner une vue différente de la vue de base.

Dans ce cas précis, sachant que j'avais deux sections dotées d'une pagination et d'appel en Ajax, j'ai dû détecter quelle clé se trouvait dans ma requête pour savoir si je voulais afficher des matchs ou des résultats et ensuite retourner la bonne vue avec les bonnes données.

*Code gérant la valeur de la réponse de la requête Ajax au sein de mon contrôleur*

```

115 // ***** Ajax *****
116 if ($request->ajax()) {
117     $input = $request->input();
118     if (array_key_exists ('games', $input)) {
119         return view('ajax.games', ['games' => $games])->render();
120     } else if (array_key_exists ('results', $input)) {
121         return view('ajax.results', ['results' => $results])->render();
122     }
123 }
```

Cette mise en place est très simple grâce au système de pagination intégré à Laravel. Elle est très facile à prendre en main et ne demande pas beaucoup de code... encore une raison d'aimer Laravel.

Dans ma page d'accueil et mon calendrier je souhaitais implémenter un système de filtre pour les matchs et les résultats. Dans ces deux pages, les matchs et résultats sont tous regroupés indépendamment de leur équipe et sont triés par date du plus proche au plus éloigné. Sachant que j'avais décidé de n'afficher que 6 matchs et 3 résultats par page, je souhaitais donner la possibilité aux utilisateurs de filtrer le contenu de cette page et de ne consulter que les matchs qui les intéressent. Dans la même perspective de performance, je souhaitais que cette modification de contenu soit faite en Ajax pour éviter de charger la page entière à chaque filtre. Je me suis donc replongé dans l'Ajax mais cette fois avec un peu plus de difficultés. Dans un premier temps, je devais créer mes filtres. Pour les équipes, rien de plus simple, mais pour le calendrier je ne pouvais pas le coder entièrement moi-même, j'ai donc été à la recherche d'un plugin qui me générerait un calendrier. J'ai trouvé « monthly » un plugin que Bernard avait déjà utilisé pour un de ses projets chez Exit Studio et qu'il m'avait conseillé. Une fois mes filtres mis en place, je devais stocker la sélection de mes filtres avant de les appliquer.

*Stockage des filtres avant de les appliquer*

```

8   // ***** Get data from team filters *****
9
10  var teamFilters = new Array();
11
12  $('body').on('click', '.filters-teams-team', function(elt){
13
14    teamFilters.length = 0;
15
16    $(this).toggleClass('selected');
17
18    $('.filters-teams-team.selected').each(function(elt){
19
20      team = $(this).attr('data-teamID');
21
22      teamFilters.push(team);
23
24    })
25
26  });

```

```

18  // ***** Get data from date filters *****
19
20  var dateFilters = new Array();
21
22  $('body').on('click','.monthly-day-pick', function(elt){
23
24    if (!$(this).hasClass('monthly-past-day')) {
25
26      $(this).toggleClass('selected');
27
28    } else {
29
30      $(this).removeClass('selected');
31
32    }
33
34    date = $(this).data('date');
35
36    // ***** Check if the date is already in the array *****
37
38    dateCheck = $.inArray(date, dateFilters);
39
40    if (dateCheck == -1) {
41
42      // ***** store data in array *****
43
44      dateFilters.push(date);
45
46    } else {
47
48      // ***** remove data from array *****
49
50      if (!Array.prototype.remove) {
51
52        Array.prototype.remove = function(val) {
53
54          var i = this.indexOf(val);
55
56          return i>-1 ? this.splice(i, 1) : [];
57
58        };
59
60      }
61
62      dateFilters.remove(date);
63
64    }
65
66    window.history.pushState('', '', '/?dates=' + dateFilters);
67
68  });

```

Une fois mes filtres stockés, il ne me restait plus qu'à les appliquer à mes données et les retourner en Ajax sous forme de HTML. Pour que cela fonctionne j'ai décidé de dédier une url particulière lorsqu'un filtre est appliqué, ce qui m'a permis d'avoir un contrôleur spécifique pour cette requête Ajax.

*Fonction spécifique du contrôleur pour filtrer des matchs de la page d'accueil*

```

66     public function homeFilters(Request $request)
67     {
68         $data = $request->data;
69
70         // ***** Get teams ID *****
71
72         if (isset($data['teams'])) {
73
74             $teamsID = $data['teams'];
75
76         }
77
78         // ***** Get teams ID *****
79
80         if (isset($data['dates'])) {
81
82             $dates = $data['dates'];
83
84         }
85
86         // ***** Get filtered matchs *****
87
88         $game = new Game;
89
90         if (isset($dates) && isset($teamsID)) {
91
92             $games = $game -> getFilteredGames($dates, $teamsID);
93
94         } else if (!isset($dates) && isset($teamsID)) {
95
96             $games = $game -> getFilteredGames(null, $teamsID);
97
98         } else if (isset($dates) && !isset($teamsID)) {
99
100            $games = $game -> getFilteredGames($dates, null);
101
102        }
103
104        $view = View::make('ajax.games') -> with('games', $games) -> render();
105
106
107        return response()->json( array('success' => true, 'html' => $view) );
108
109    }

```

Concernant la fonction Ajax, j'ai dû donner quelques informations supplémentaires par rapport à l'Ajax de mes paginations. J'ai dû structurer les « data » pour recevoir les filtres des équipes ainsi que ceux des dates. Pour que ma requête soit autorisée, j'ai dû récupérer le token généré par Laravel. J'ai ensuite dû préciser le type de la méthode, et comme pour les paginations, l'url utilisée.

*Fonction Ajax pour appliquer les filtres aux matchs de ma page d'accueil*

```

45 // ***** Manage ajax *****
46
47     $('body').on('click', '.filters-submit', function(e) {
48
49         e.preventDefault();
50
51         var data = { teams : teamFilters, dates : dateFilters };
52
53         url = $(this).attr('href'),
54
55         token = $("meta[name='csrf-token']").attr('content'),
56
57         teamFiltersLength = teamFilters.length,
58
59         dateFiltersLength = dateFilters.length;
60
61         if (data == null) {
62
63             $(this).after('<div class="feedback-error"><p>Choisissez des filters avant de valider...</p></div>');
64
65         } else {
66
67             $.ajax({
68
69                 type: "POST",
70
71                 url: url,
72
73                 data: { '_token': token, 'data': data },
74
75                 success: function(response) {
76
77                     $('.gameCard-wrap').html(response.html);
78
79                 }
80             });
81
82         }
83     });
84
85 });

```

Je devais bien entendu gérer certains cas particuliers, lorsqu'il n'y avait que des filtres par rapport aux équipes ou, à l'inverse, quand il n'y en avait que par rapport aux dates. J'ai dû anticiper la possibilité de ne recevoir aucune donnée à afficher, et dans ce cas rendre un feedback à l'utilisateur. J'ai ensuite transféré ce que j'avais fait pour les filtres des matchs de la page d'accueil aux résultats de cette même page et de même sur la page du calendrier.

## Conclusion

Il est intéressant pour moi de prendre un peu de recul par rapport à ce projet sur le point d'aboutir. Ce fut en effet le défi le plus concret que j'aie relevé jusqu'à présent, autant pour son côté formateur que pour l'investissement personnel que j'y ai consenti.

Concernant son côté formateur, ce projet m'a appris énormément en terme de professionnalisme et de contact avec le client. Déchiffrer les attentes d'un client ou imposer mon avis professionnel sur certains points représentait en effet pour moi un nouvel exercice. J'ai bien sûr développé mes compétences techniques dans les domaines que je connaissais déjà tels que le javascript ou les requêtes Ajax grâce à une envie et un besoin de dépasser mes limites. J'ai acquis une meilleure maîtrise de l'architecture « Model-View-Controller » qu'utilise Laravel. Ce projet m'a également obligé à relever le challenge d'apprendre par mes propres moyens à utiliser un nouveau CMS. Je suis fier d'avoir étendu mes compétences avec Laravel. Ce challenge m'a permis de prendre confiance en mes capacités.

Concernant mon investissement personnel, je n'ai jamais considéré ce travail comme un projet scolaire. Étant en directe relation avec le client, et concerné personnellement par l'aboutissement de ce projet, c'était pour moi très important. Ce site est bientôt en ligne et je prévois de le généraliser en vue de sa réutilisation pour d'autres clubs. Certains dirigeants m'ont déjà manifesté leur intérêt.

Bien que mon travail touche à son aboutissement, il n'est pas pour autant terminé définitivement. En effet, j'envisage une poursuite de ma collaboration avec les responsables de mon club pour ajouter toute une série de nouvelles fonctionnalités. De plus je ne considère pas cette réalisation comme un « one shoot » mais plutôt comme un projet qui me suivra et évoluera tout au long de ma carrière. Je suis cependant assez fier d'en présenter la version 1.0.

Pour moi, ce travail symbolise ma transition de la vie d'étudiant à la vie professionnelle, un tremplin idéal.

## Bibliographie

### Liens utilisés pour ce projet :

- Documentation PHP  
<http://php.net/docs.php>
- Documentation Laravel  
<https://laravel.com/docs/5.4>
- Documentation Backpack  
<https://laravel-backpack.readme.io/docs/getting-started>
- Documentation jQuery  
<http://api.jquery.com/>
- Plugin Accessible Mega Menu  
<https://adobe-accessibility.github.io/Accessible-Mega-Menu>
- Documentation Google  
<https://developers.google.com/maps/documentation/javascript/adding-a-google-map>
- Plugin Masonry  
<https://github.com/desandro/masonry>
- Plugin Slick  
<http://kenwheeler.github.io/slick/>
- Plugin Lightbox  
<https://github.com/lokesh/lightbox2/>
- Plugin Waypoint  
<https://github.com/imakewebthings/waypoints>
- Plugin Monthly  
<https://github.com/kthornbloom/Monthly>
- Github de mon projet  
<https://github.com/Mucht/PFE>

## Remerciements

Pour la réalisation de ce travail, je tiens à remercier tout particulièrement Madame Myriam Dupont, promotrice de mon travail de fin d'études pour m'avoir épaulé tout au long de ce projet.

Je remercie particulièrement Jean-Paul Houart, secrétaire du RBC Ciney, pour sa collaboration et sans qui ce projet n'aurait pas vu le jour.

Pour leur soutien moral, leurs précieux conseils et leur disponibilité, j'adresse mes vifs remerciements également aux personnes ressources :

- Monsieur Dominique Vilain, professeur,
- Monsieur Bernard Petithan, maître de stage,
- Monsieur Julien de Hosté, maître de stage,
- Monsieur Maxime Dessambre, étudiant en informatique.

Merci encore à :

- mon papa, pour la relecture de mon rapport,
- ma maman qui, par sa grande vigilance, m'a permis de rester concentré sur mon objectif,
- Manon Duvivier pour sa contribution aux petites touches d'humour,
- Anne Laure, Axel, Frédéric et les autres qui m'ont tendu le verre au moment où j'en avais le plus besoin.