

2021

DOSSIER DE PROJET

Refonte du site de l'association Handball
Club de Pontchâteau



Andreani Quentin
Promotion 2020 - 2021
Développeur Web et Web Mobile
ARINFO - Saint-Nazaire

Avant - Propos	4
Remerciement	4
Listes des compétences couvertes par le référentiel	5
Environnement de travail	6
Résumé du projet	6
Les outils utilisés	7
Veille	9
Cahier des charges	10
Expression des besoins	10
Spécifications Fonctionnelles	11
Arborescence détaillée	12
Code Couleur	13
Logo	14
Police d'écriture	14
Maquetter une application	15
Réalisation du site	16
Planification du projet	16
Définition des rôles	17
MCD / MLD et Diagrammes UML	18
MCD	18
MLD	19
Use Case Diagramme	19
Diagrammes d'activités	20
Développement back-end	21
Création du squelette et de la base du projet	21
Construction de la base de donnée	21
Création des entités	22
Utilisateur et sécurité	23
Veille sur la sécurité	25
Exemple de code	29
Le gestionnaire d'image	29

Réaliser une interface utilisateur avec gestion de contenu	34
Front-end	40
Mise en place du Framework CSS	40
Réaliser une interface web statique et dynamique	42
Mise en production	46
Le référencement	48
Test de performance	51
Recherche en anglais	52
Vie privée	54
Conclusion	56

AVANT - PROPOS

Ce dossier professionnel a été réalisé dans le cadre d'une reconversion professionnel. J'ai bénéficié de l'aide de l'état CPF afin de réaliser cette formation que j'ai commencé le 28 Septembre 2020. J'ai toujours été un passionné d'informatique mais ayant décroché les études un peu tôt pour travailler en restauration, il était évident pour moi de reprendre un jour mes études afin de réaliser quelque chose qui me plaisait vraiment.

J'ai eu une petite expérience du code via **l'école 42**, j'ai été en immersion pendant un mois dans la « **Piscine** », une étape où l'on apprend à coder en « **C** ». Cette expérience a été très enrichissante pour moi et m'a conforté dans l'idée d'une reprise de mes études dans le domaine du développement web.

J'ai très vite compris, via cette petite expérience, toutes les possibilités offertes par le développement et la programmation en général, et le potentiel illimité d'apprentissage que nous offres toutes les documentations et ressources disponibles gratuitement, afin d'appréhender les bases des différents langages et technologies que nous pouvons utiliser.

La formation s'est déroulée en partie en télétravail avec une phase de confinement suite à la conjoncture actuelle, cependant cela n'a aucunement affecté la qualité de l'enseignement, ma motivation et ma détermination.

Remerciement

Je tiens tout d'abord à fortement remercier Nicolas Gicquel à la fois mon ancien coach et mon formateur, pour la qualité de sa formation, sa disponibilité, sa pédagogie et surtout son sens de l'humour à toutes épreuves.

Je remercie également Anaïs Limousin pour ses conseils en WebDesign qui ont été précieux, Elodie Boinet directrice du centre Arinfo de Saint-Nazaire pour son soutien qui m'a permis de réaliser cette formation, ainsi que Marion Lamare qui m'a donné de précieux conseils pour mon CV et ma recherche de stage que j'ai trouvé grâce à elle.

Enfin je remercie mes camarades de promo, Emmanuel, Alyssa, Nicolas et Alain, pour l'entraide bienveillante dont ils ont tous fait preuve.

Listes des compétences couvertes par le référentiel

1. Maquetter une application.
2. Réaliser une interface utilisateur web statique et adaptable.
3. Développer une interface utilisateur web dynamique.
4. Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce.
5. Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité.
6. Créer une base de données.
7. Développer les composants d'accès aux données.
8. Développer la partie back-end d'une application web et web mobile.
9. Elaborer et mettre en oeuvre des composants dans une application de gestion de contenu ou e-commerce.

ENVIRONNEMENT DE TRAVAIL

Résumé du projet

L'association Handball Club de Pontchâteau existant depuis 2012, souhaitait avoir un nouveau site vitrine dans le but d'apporter un nouveau support de communication et accessible au plus grand nombre. Des discussions ont été entamées afin d'évaluer la charte graphique ainsi que les différents éléments qui devront y être présents.

Concernant les requêtes sur la charte graphique émises par l'association, le site devra avoir une interface simple et accessible. Il devra contenir un diaporama en page d'accueil et le rouge, blanc et noir seront les couleurs dominantes sur tout le site.

J'ai aussi proposé une refonte du logo de l'association pour quelque chose de plus moderne.

Après réflexion, nous avons pu établir sur les grandes lignes l'aspect fonctionnel du site. Partant de ses éléments j'ai pu établir un cahier des charges plus élaborés côté développement. Une veille concurrentielle a scrupuleusement été faite afin d'apporter des idées et des solutions pendant toute la rédaction du cahier des charges pour que le projet corresponde au mieux aux attentes.

Des wireframes ont été créés afin de valider la charte graphique du site en mode desktop et mobile et des diagrammes de séquences, UML ainsi qu'un MCD/MLD ont été réalisés, amenant un visuel sur les relations et le fonctionnement du site.

Élaboration d'une arborescence détaillée, comprenant un détail global des différentes pages avec pour chacune d'entre elles leurs contenus.

Pour la partie confidentialité, des mentions légales seront mise en place ainsi qu'une politique de confidentialité couplé à une gestion des cookies.

Le tout sera élaboré via les technologies PHP, Symfony et MySQL pour la partie Back-End du site. Ainsi que les technologies HTML/CSS/ Sass, Bootstrap et Javascript/JQuery pour la partie Front-End du site.

L'association m'a laissé beaucoup de liberté, ce qui m'a permis une créativité plus présente, je prendrais le temps de m'étendre sur le sujet dans la « Conclusion » à la fin de mon dossier.

Les outils utilisés

Voici les différents outils, logiciel, Frameworks utilisés pour le développement du site.

BACK-END :

- ▶ Pour l'environnement du serveur local j'ai utilisé « **MAMP** » qui m'a permis de tester et de développer mes scripts **PHP**. J'ai longuement hésité avec le logiciel « **XAMPP** » mais j'ai constaté quelques problèmes avec ce dernier, ce qui m'a conforté à utilisé « **MAMP** » qui pour moi est le meilleur logiciel environnement de serveur local pour les utilisateurs de Mac comme moi.
- ▶ Pour le plus gros morceau j'ai utilisé « **Symfony** » qui est un Framework Open-Source MVC (Modèle - Vue - Contrôleur) basé sous le langage **PHP** (Hypertext Preprocessor).
- ▶ En ce qui concerne la base de données j'ai utilisé « **MySQL** » (My Structured Query Language) qui est un gestionnaire Open-Source couplé à « **PHP My Admin** » une interface simple et facile d'accès.

FRONT-END :

- ▶ Tout d'abord j'ai utilisé le préprocesseur **CSS** (Cascading Style Sheet) « **SASS** » (Syntactycally Awesome Stylesheets). Il permet de mieux gérer son code CSS et facilite les mises à jour en cas de modification du code par exemple avec des **Mixins** ou des **Extends**.
- ▶ Ensuite j'ai utilisé le Framework « **Bootstrap** » que j'apprécie tout particulièrement. Bootstrap est une collection d'outils utiles à la création du design de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. Son Système de Grid est particulièrement efficace.
- ▶ Pour les effets dynamiques du site j'ai utilisé le langage « **JavaScript** » et sa bibliothèque « **JQuery** ».

LOGICIELS :

- ▶ Pour mon environnement de développement j'ai privilégié l'éditeur de texte « **Sublime Text** » que je trouvais bien plus épuré que l'IDE « **VS Code** ». Je trouve, pour ma part, que Sublime Text n'a rien à envier à Visual Studio Code, ne serais-ce que par sa grosse communauté, son environnement épuré et la montagne de plugin et package que nous pouvons lui attribuer.
- ▶ En ce qui concerne le terminal j'utilise « **iTerm** » qui est une excellente alternative au terminal natif de Mac OS, je le couple bien évidemment au gestionnaire de paquet indispensable « **Homebrew** » ainsi que le gestionnaire de dépendance « **Composer** ». Ils m'ont permis la construction de mon projet, le téléchargement des différents bundles ainsi que d'effectuer les mises à jour nécessaires.
- ▶ J'ai synchronisé mon projet sur « **GitHub** » afin de faciliter la programmation avec la permission de revenir en arrière en cas de problème dans mon code. Cela m'a permis durant mon développement de me sauver de quelques sueurs froides suite à la perte de certaines données (Problème avec l'environnement XAMPP notamment). GitHub offre aussi la possibilité de partager son code sur plusieurs ordinateurs avec un système de « clone » ce qui m'a bien servi durant tout la phase de développement de mon projet et ma facilité l'échange avec mon formateur en cas de problème qui pouvait y accéder à n'importe quel moment.

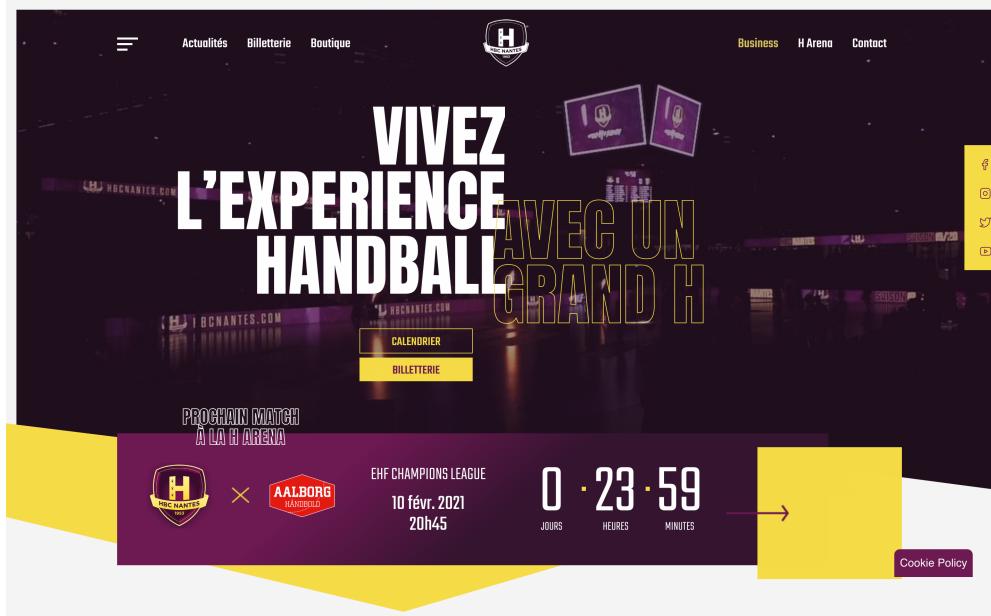
Veille

Afin d'établir une refonte du site web de l'association « PCHB » (PontChâteau Handball Club), et de correspondre au mieux à leurs attentes, j'ai effectué différentes recherches afin de trouver des renseignements sur divers sites de sport.

Mes recherches m'ont beaucoup aidé à trouver des solutions design et efficace pour répondre aux besoins de l'association, j'ai découvert des sites très intuitifs et intéressant.

Voici les différents sites qui m'ont beaucoup aidé pour le design et les fonctionnalités que les sites de sports ont tendances à utiliser :

- ▶ Football Club de Nantes : www.fcnantes.com
- ▶ Les Girondins de Bordeaux : www.girondins.com
- ▶ Le Stade Nantais Rugby : www.stadenantais.fr
- ▶ Le Handball Club de Angers : www.handball.angers-sco.fr
- ▶ Le Handball Club de Nantes : www.hbcnantes.com



Voici le site du HBCN (HandBall Club Nantais) qui a fortement inspiré le design final du projet.

The website layout includes two main sections below the header: "LES ACTUALITÉS" (News) and "LES BRÈVES" (Highlights). The "Actualités" section features a video thumbnail of a man speaking, a date (9 FÉVRIER 2021), and a title ("Grégory Cojean: 'Une rencontre de très haut niveau'"). The "Brèves" section lists several news items with dates and titles, such as "Adria Figueras dans le 7 de la semaine" (9 FÉVRIER 2021 19:49) and "Le programme de la semaine" (8 FÉVRIER 2021 11:44).

CAHIER DES CHARGES

Expression des besoins

L'ancien site étant vieillissant, l'association avait besoin d'un site plus moderne, notamment pour se faire connaître et améliorer leur visibilité. Le site devra reprendre les principaux axes de l'ancien site, et être modulable via une gestion de contenu.

PRINCIPALES DEMANDES :

- ▶ L'inscription de nouveaux administrateur
- ▶ Mettre en place un formulaire de contact
- ▶ Incorporer un système de planning pour les entraînements
- ▶ Mettre en avant les résultats du club ainsi que le calendrier des matchs
- ▶ Mettre un système d'administration, permettant la gestion des contenus
- ▶ Incorporer un système de login mot de passe pour l'accès utilisateur et administrateur
- ▶ Réaliser une page d'information pour les inscriptions au club avec tout les documents relatifs à l'inscription

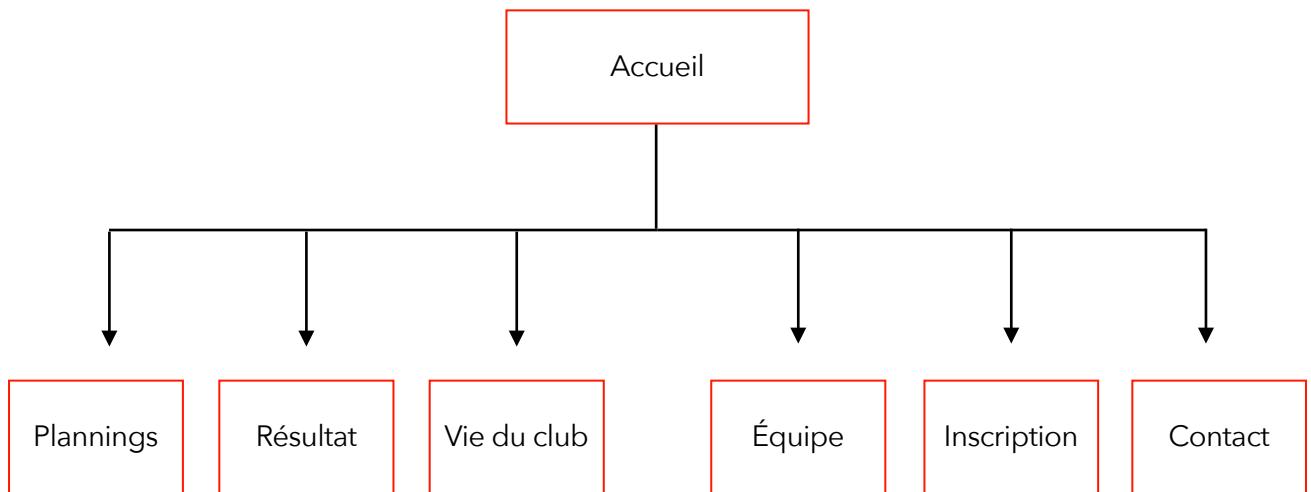
POUR LA PARTIE VISUELLE DU SITE :

- ▶ La page d'accueil devra comporter un carrousel d'images
- ▶ Le site aura pour couleur dominante le rouge, le noir et le blanc
- ▶ Il comportera un slider pour la partie sponsors
- ▶ Il contiendra un parallaxe sur la page principale
- ▶ Le logo du club devra être mis en avant sur la bar de navigation en version
- ▶ Le logo du club sera éventuellement retravaillé

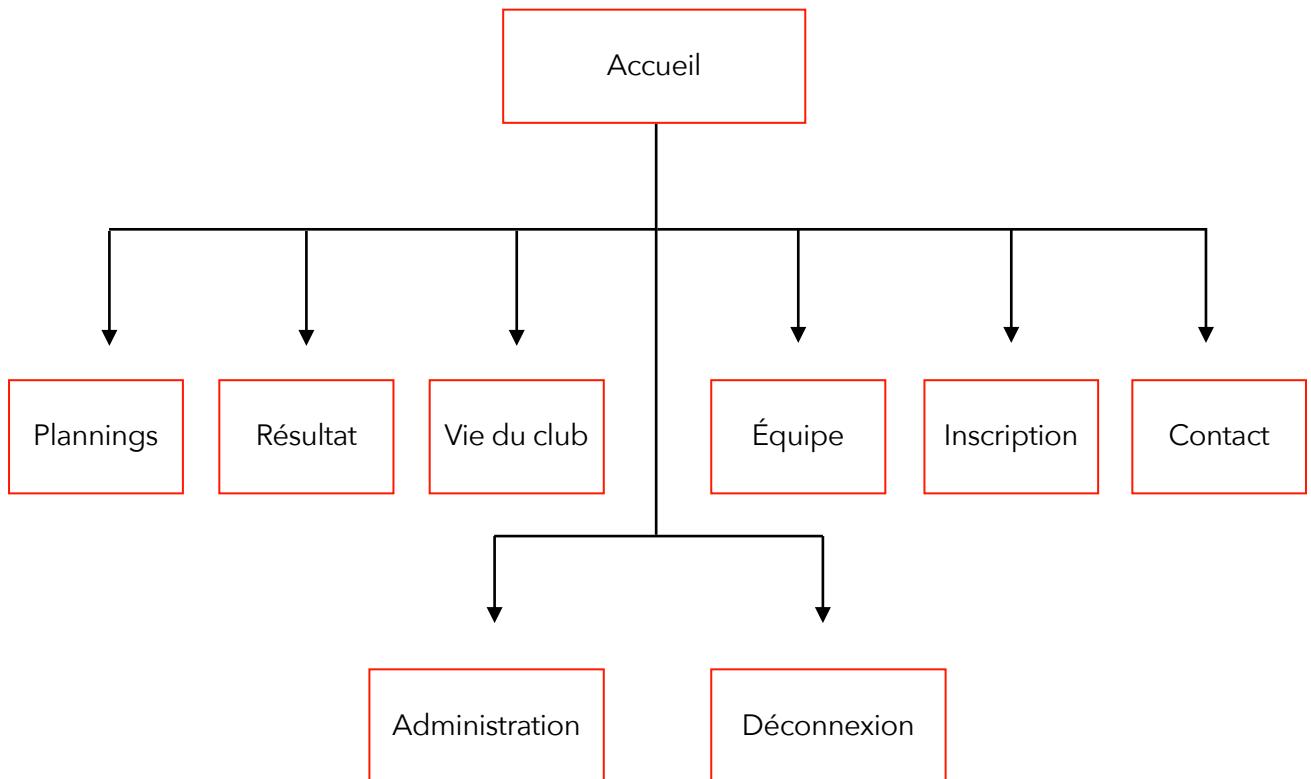
Spécifications Fonctionnelles

Voici l'arborescence du site en fonction des différents statuts :

Accès Visiteur



Accès Administrateur



Arborescence détaillée

Chaque page du site contient plusieurs éléments dont voici les principaux :

ACCUEIL :

- ▶ Titre bien visible de l'association avec un bouton vers le calendrier des matchs
- ▶ Carousel d'images personnalisable
- ▶ Iframe Scor'n'Co des prochains match et résultats des matchs de l'équipe principal et iframe Facebook
- ▶ Les trois derniers articles d'actualités du club afin d'informer les visiteurs
- ▶ Une rubriques communauté avec un mot du président et les liens vers les réseaux sociaux
- ▶ Une section sponsors, afin d'afficher les partenaires du club dans un slider automatique

PLANNINGS :

- ▶ Iframe Google Agenda permettant aux visiteurs de consulter le planning des entraînements et les éventuelles événements qui peuvent être planifiés.

RÉSULTATS :

- ▶ Iframe Scor'n'co permettant de visualiser les résultats des matchs de toutes les équipes ainsi que leurs calendriers des matchs à venir.

VIE DU CLUB :

Actualités du club :

- ▶ Page contenant les articles d'actualités du club avec chacun une image dédié.
- ▶ L'article le plus récent est en tête de page.

Organigramme :

- ▶ Page contenant l'organigramme du club, une simple image réalisé par le club.

Historique :

- ▶ Page contenant l'historique du club, modifiable par un administrateur.

ÉQUIPES :

- ▶ Les pages contiennent une photo de l'équipe en question.
 - ▶ Les pages contiennent une description de l'équipe et éventuellement des liens ajoutés par l'administrateur.
-

INSCRIPTIONS :

- ▶ La page Inscriptions contient un contenu personnalisé par l'administrateur du site via ck Editor
 - ▶ Les documents PDF téléchargeable pour l'inscription des mineurs.
 - ▶ Les documents PDF téléchargeable pour l'inscription des majeurs.
 - ▶ Les documents PDF téléchargeable ressources.
 - ▶ La liste des tarifs.
-

CONTACT :

- ▶ Page permettant de contacter le club pour tout types de renseignements.
 - ▶ La page contient aussi l'adresse du gymnase où exerce le club et une implémentation d'une carte Open Street Map.
-

ADMINISTRATION :

- ▶ Possibilité de modifier toutes les pages du site
- ▶ Gestion des utilisateurs, notamment l'ajout d'un nouvel Administrateur
- ▶ Accès à la messagerie de la page contact

Code Couleur

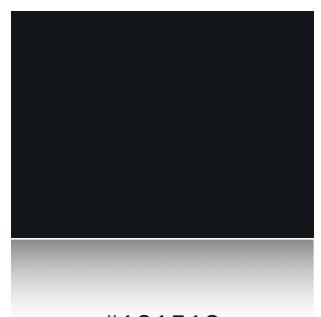
Par soucis de correspondance avec les couleurs originels du club qui sont le noir, le rouge et éventuellement le blanc voici les couleurs qui ont été retenues :



#FF0100



#2B2F36



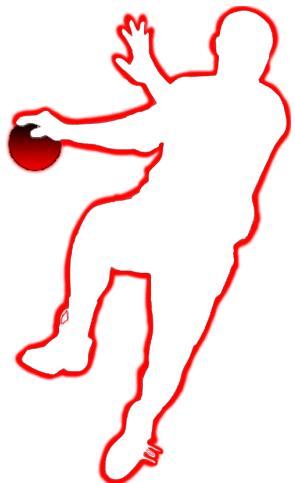
#121519

Logo



Le club possède un logo, il s'agit du même depuis de nombreuses années, je me suis donc proposé pour le rafraîchir un peu afin de le rendre transparent et un peu plus visible sur différents arrières plan.

J'ai donc réalisé un détourage et rendu transparent l'intérieur, le tout avec Photoshop.



Police d'écriture

Deux polices ont été définies pour tout le site, les polices ont été ajouté dans le projet via Google Fonts.

Les titres seront écrits avec la police **Anton**, elle correspond bien à l'idée du caractère sportif du club.

Penultimate
The spirit is willing but the flesh is weak
SCHADENFREUDE
3964 Elm Street and 1370 Rt. 21
The left hand does not know what the right hand is doing.
mail@example.com http://www.cufonfonts.com

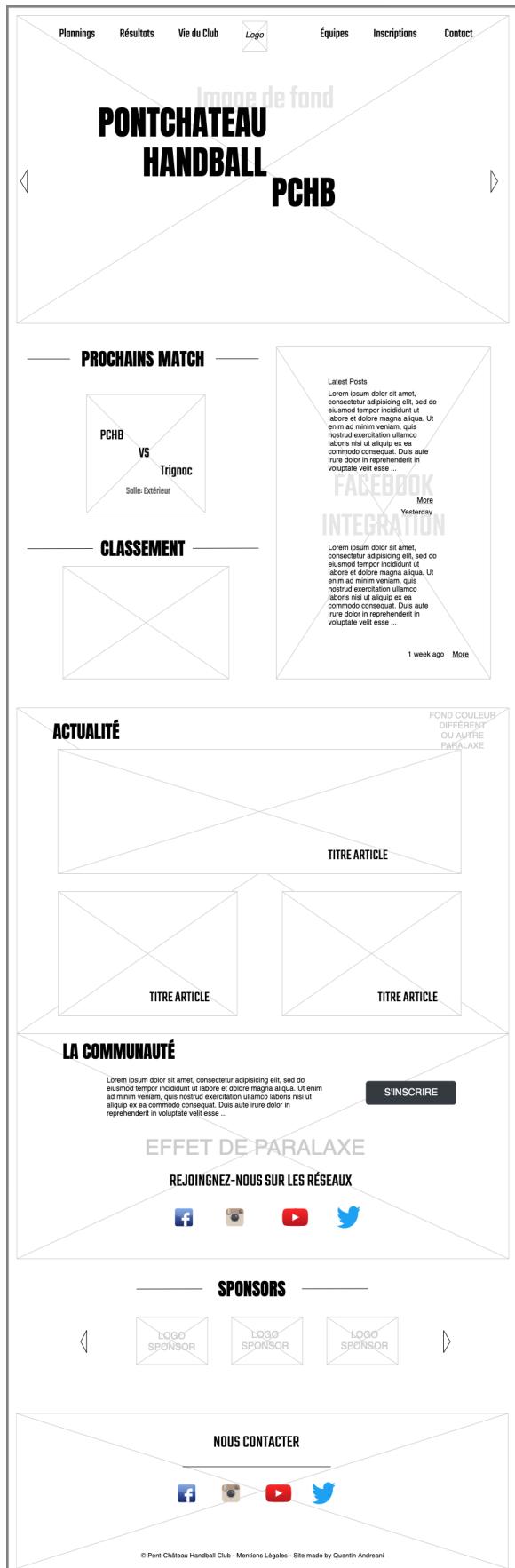
Penultimate
The spirit is willing but the flesh is weak
SCHADENFREUDE
3964 Elm Street and 1370 Rt. 21
The left hand does not know what the right hand is doing.

Quelques autres textes ainsi que les boutons utilisent la font **Teko**.

Une police plutôt populaire pour ce genre d'utilisation.

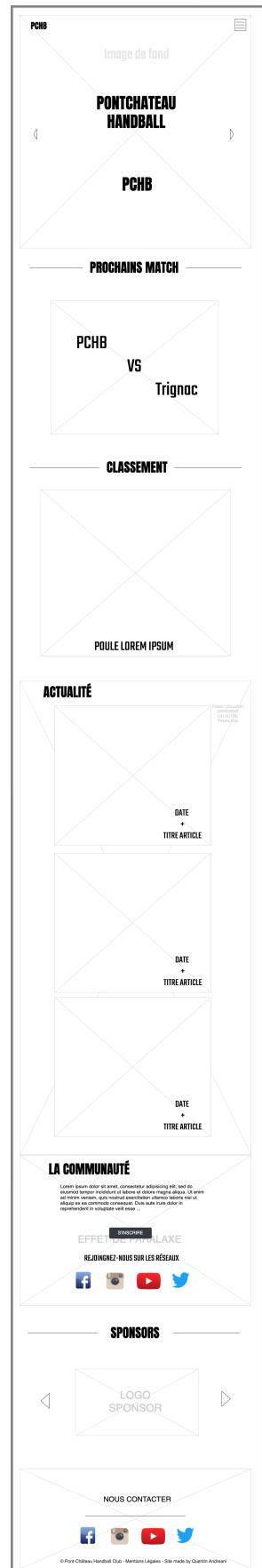
Pour le reste des rubriques c'est au choix pour le club via le bundle CK Editor.

MAQUETTER UNE APPLICATION



Afin de réaliser une maquette du site du PCHB, j'ai utilisé l'application gratuite **Draw.IO** pour créer des wireframes.

À gauche nous avons la wireframe de la page d'accueil.



À droite nous avons la wireframe de la page d'accueil responsive.

RÉALISATION DU SITE

Planification du projet

J'ai réalisé une liste de tâches détaillé dans l'ordre chronologique.

Je tiens tout de même à notifier qu'il m'aurait été possible de réaliser un diagramme de **Gant**, cependant, ne connaissant pas encore tout à fait le Framework Symfony j'étais dans l'incapacité de quantifier correctement mon temps utilisé pour les différentes parties du site.

BACK-END :

- ▶ Définir les rôles utilisateur.
- ▶ Création de diagrammes.
- ▶ Création du squelette Symfony.
- ▶ Création de la base de données.
- ▶ Création des entités avec leurs relations.
- ▶ Mise en place du service global d'authentification utilisateur.
- ▶ Sécurité du site et utilisateur.
- ▶ Mise en place de la gestion de cookies, reCAPTCHA
- ▶ Test de sécurité

FRONT-END :

- ▶ Mise en place du framework CSS Bootstrap
- ▶ Travail sur la partie Header
- ▶ Travail sur la partie Footer
- ▶ Intégration des différents blocs en fonction des pages
- ▶ Travail sur le responsive design

FINALISATION :

- ▶ Achat du nom de domaine
- ▶ Passage en production
- ▶ Phase de test
- ▶ Correction des bugs
- ▶ Test du site en version mobile
- ▶ Livraison du site

Définition des rôles

Pour ce qui concerne les rôles, j'ai fait le choix simple de prendre en compte uniquement deux rôles. En effet le site ne contenant pas de système d'inscription pour les visiteurs, je n'ai alors besoin que d'un seul rôle utilisateur et un rôle admin. Je précise aussi que le club n'a pas le besoin de plusieurs rôles différents, chaque utilisateur sera un membre du bureau, ainsi ils seront en mesure de modifier les contenus du site.

L'UTILISATEUR

L'utilisateur est le « User », il pourra accéder à l'espace Administration, modifier les différentes pages du site, ajouter ou supprimer un autre « User ».

L'ADMINISTRATEUR

Dans le cas de ce site, l'Administrateur n'est autre que le Super-Administrateur, c'est à dire le développeur. Il aura accès à la totalité du site et la possibilité de le modifier. Il aura accès au serveur à la base de données ainsi qu'au code source du site.

Le site étant encore en développement, ce choix n'est pas immuable, et il est possible que je rajoute d'autre rôle dans une futur mise à jour.

MCD / MLD ET DIAGRAMMES UML

Ayant un formateur très minutieux et pédagogique j'ai appris à réaliser des diagrammes très intéressant afin de me donner une forme de feuille de route qui m'a beaucoup aidé dans la réalisation du site.

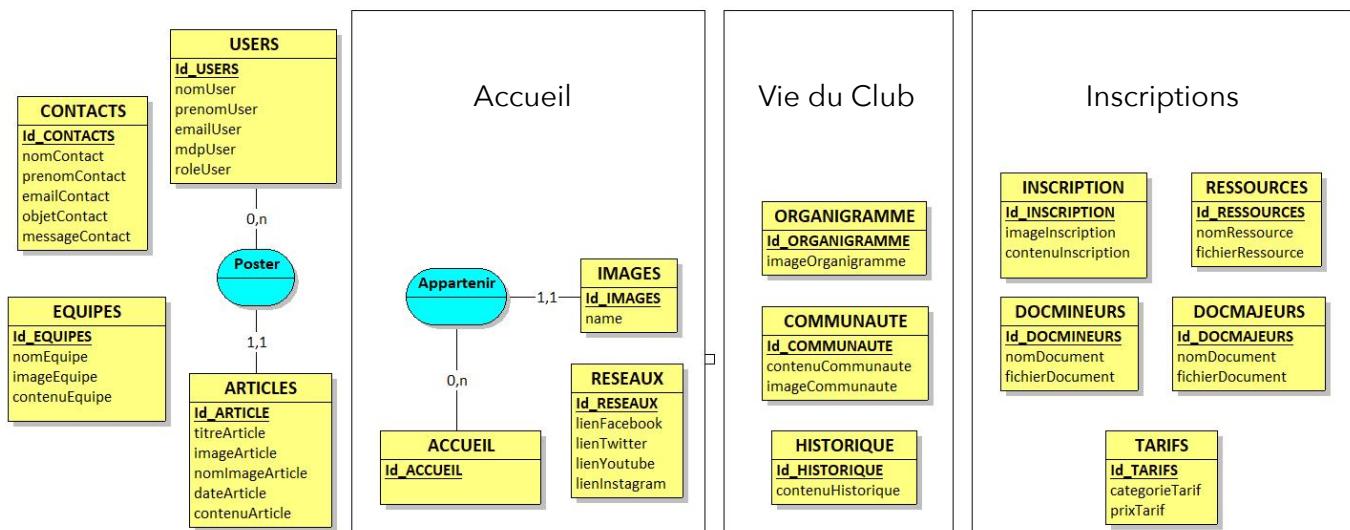
Le visuel que me donne ces éléments me permettent de mettre en avant la partie utilisateur et sécurité du site ainsi que le fait de prévoir les différents éléments que je me devais d'implémenter au niveau de mon code ainsi que des bundles que j'allais utiliser pour arriver au résultat escompté.

J'ai alors réalisé un MCD, un MLD, un Use Case Diagram, ainsi que des diagrammes d'activités.

MCD

Afin de réaliser mon MCD (Modèle Conceptuel de Données) j'ai utilisé le logiciel **Looping**, logiciel très efficace et très simple d'utilisation. Il me permet selon le modèle Entité / Relation de visualiser de manière global ma base de donnée

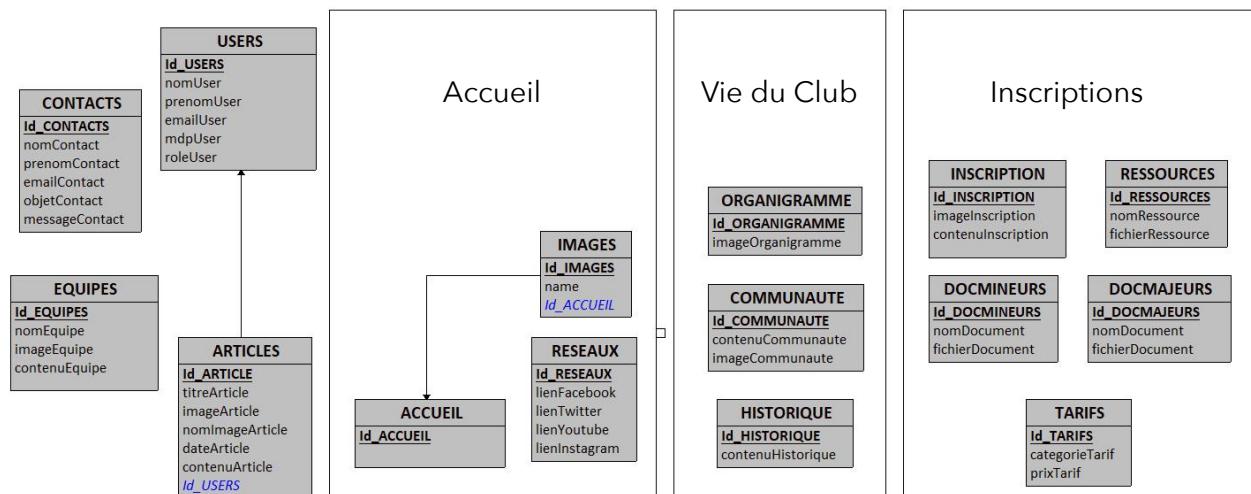
Voici le MCD que j'ai réalisé pour le site :



MLD

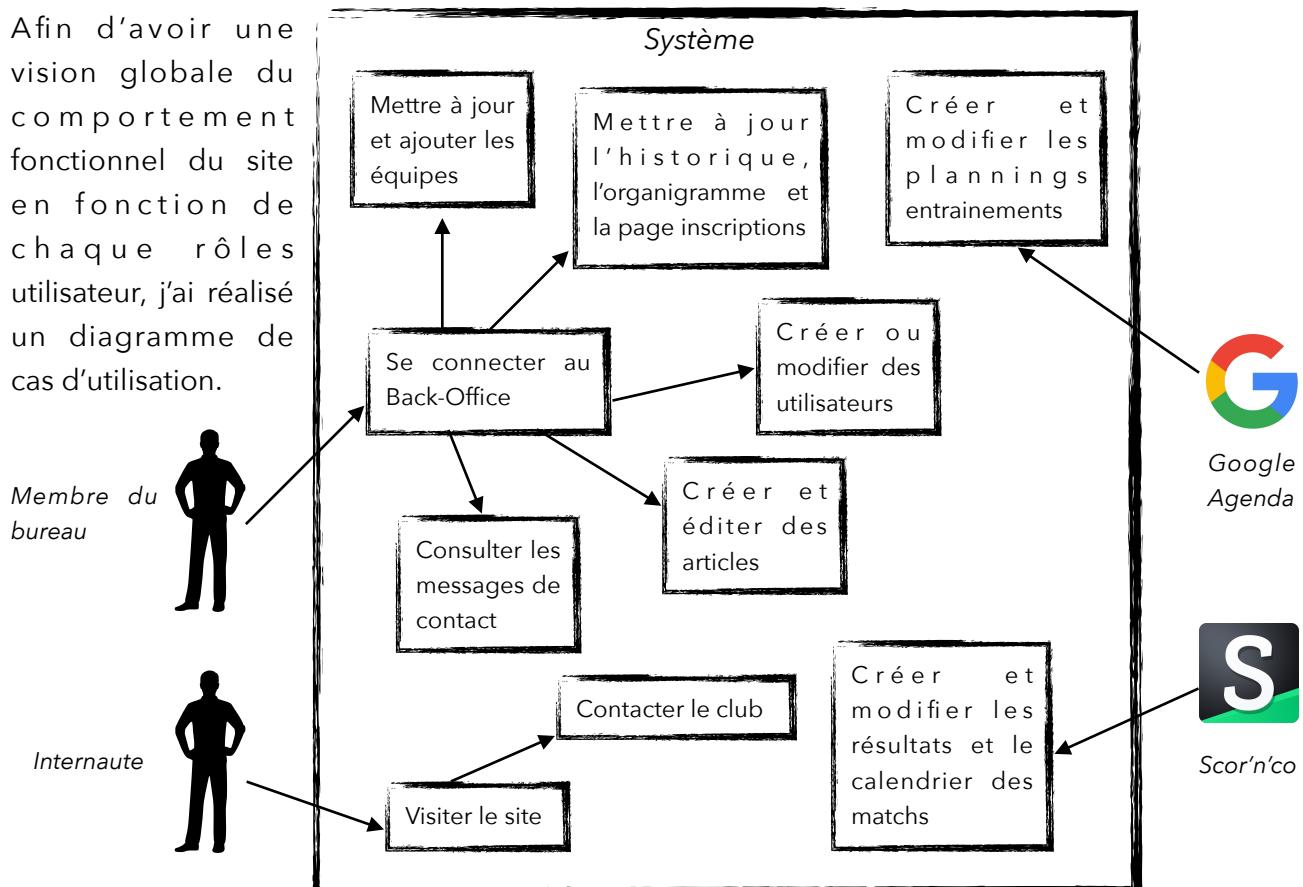
Afin de réaliser mon MLD (Modèle Logique de Données) j'ai simplement utilisé la fonction qui me permet de transformer mon MCD en MLD via Looping.

Voici le MLD :



Use Case Diagramme

Afin d'avoir une vision globale du comportement fonctionnel du site en fonction de chaque rôles utilisateur, j'ai réalisé un diagramme de cas d'utilisation.



Diagrammes d'activités

Afin de pouvoir visualiser de manière beaucoup plus précise les interactions entre usager et l'interface du site j'ai développé des diagrammes de séquence.

Voici deux de mes diagrammes d'activités :

DIAGRAMME D'ACTIVITÉ CONNEXION UTILISATEUR

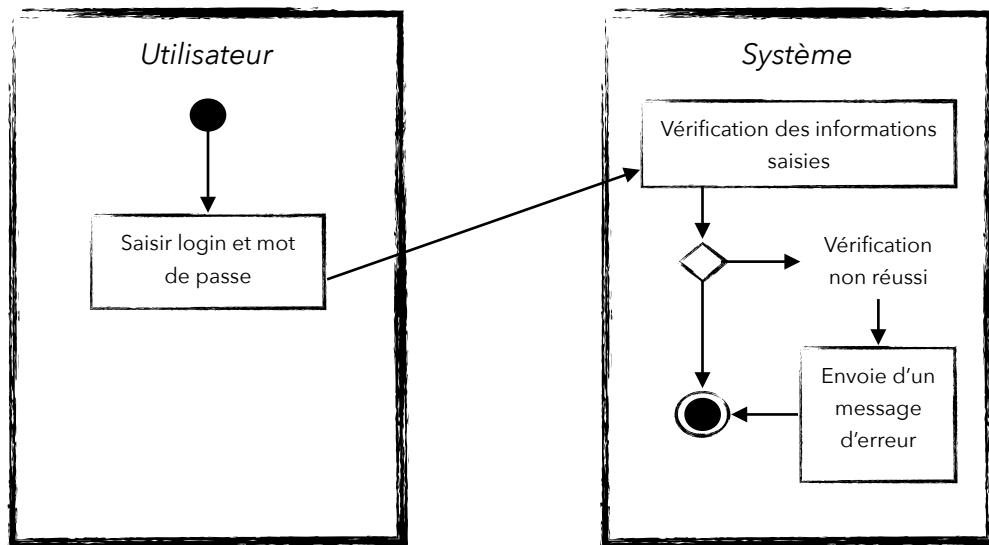
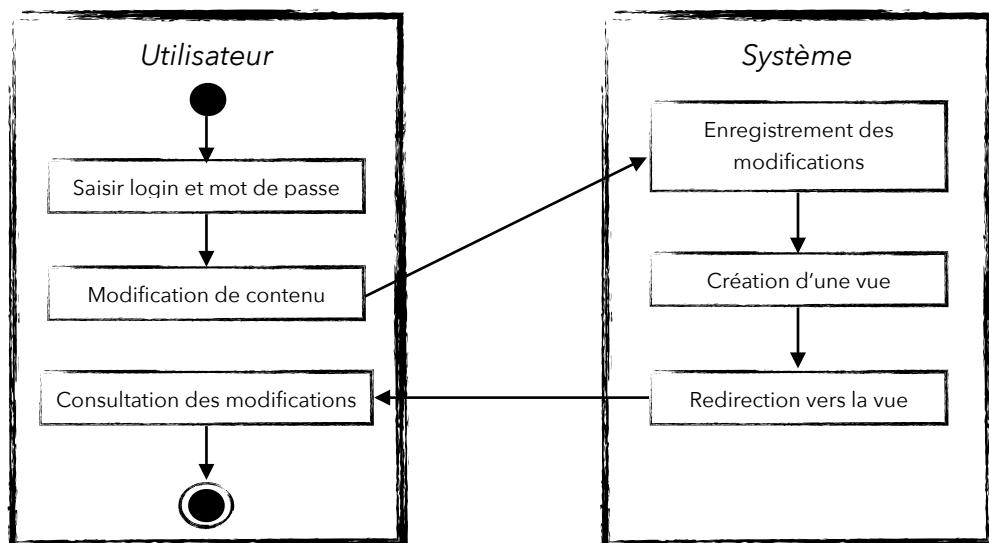


DIAGRAMME D'ACTIVITÉ MODIFICATION DE CONTENU



DÉVELOPPEMENT BACK-END

Création du squelette et de la base du projet

Pour démarrer mon projet j'ai commencé par créer un nouveau dossier de projet dans le dossier « **htdocs** » de mon serveur MAMP.

J'ai ouvert ce dernier avec mon terminal en utilisant cette ligne de commande bash :

```
→ ~ /Applications/MAMP/htdocs/projet
```

Une fois dans le dossier j'ai exécuté la ligne de commande composer afin de créer le squelette symfony pour site web :

```
→ projet composer create project symfony/website skeleton pchb
```

Construction de la base de donnée

Mon projet étant opérationnel, il me fallait maintenant mettre en place ma base de données. Pour ce faire je me devais de modifier le fichier **.env** qui se situe à la racine du projet Symfony.

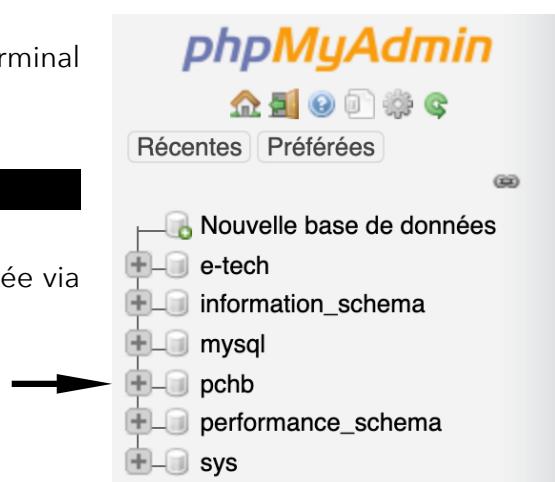
J'ai modifier la variable d'environnement « **DATABASE_URL** » afin d'obtenir une relation entre mon projet et la base de données sous MySQL.

```
29 # DATABASE_URL="sqlite:///%kernel.project_dir%/var/data.db"
30
31 DATABASE_URL="mysql://root:root@127.0.0.1:8889/pchb"
32
33 ##### doctrine/doctrine-bundle #####
34
```

J'ai ensuite entré cette ligne de commande dans le terminal afin de créer ma base de données :

```
→ pchb php bin/console database:create
```

On vérifie ensuite si la base de données a bien été créée via phpMyAdmin.



Création des entités

CRÉATION D'UNE ENTITÉ : USER

La première entité que j'ai décidé de créer est l'entité pour les utilisateurs du site : User. Elle me permettra par la suite de pouvoir générer un système de rôle, de se connecter et d'ajouter d'autres utilisateurs.

Sur Symfony, la construction des entités est très simple, elle se fait par le biais de ligne de commande dans le terminal à la racine du projet et bénéficie d'une trame chronologique nous demandant certains paramètres utiles à la création de l'entité et à sa futur utilisation, comme les données à intégrer ou encore le type de contenu.

Pour créer l'entité User j'ai utilisé la ligne de commande :

```
→ pchb php bin/console make:user
```

J'ai suivi les indications et entré les paramètres nécessaires à la création de l'entité. Une fois l'entité créée, il nous faut l'envoyer dans la base de données, cela se passe aussi en ligne de commande :

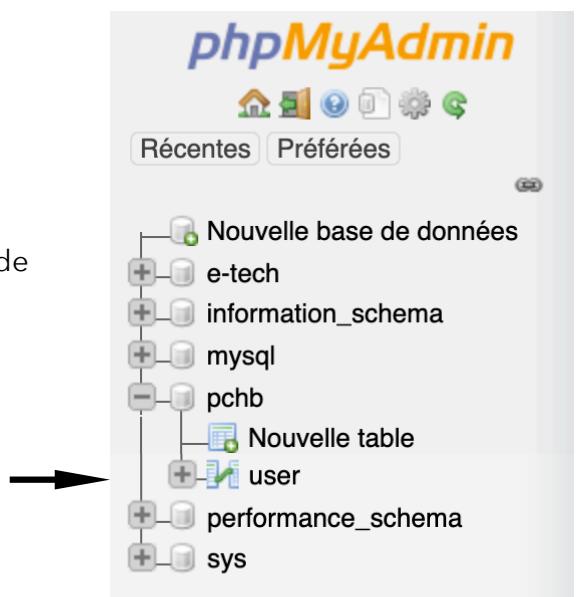
La première permet d'envoyer l'entité dans la base de données :

```
→ pchb php bin/console make:migration
```

La seconde permet de confirmer l'opération :

```
→ pchb php bin/console doctrine:migrations:migrate
```

Voilà, l'entité a été créée et envoyé dans la base de données



CONSTRUCTION DE L'AUTHENTIFICATION

Symfony nous permet de créer automatiquement un système d'authentification grâce à une ligne de commande et des paramètres à mettre en place suivant cette ligne de commande :

```
➔ pchb php bin/console make:auth
```

Cette ligne nous permet entre autre de créer un formulaire de connexion avec un mail ainsi qu'un mot de passe, j'ai par la suite ajouté deux autres paramètres dans l'entité, le nom et le prénom de l'utilisateur avec la commande :

```
➔ pchb php bin/console make:entity
```

Ce qui nous donne dans le code un formulaire comme celui-ci :

```
17     public function buildForm(FormBuilderInterface $builder, array $options)
18     {
19         $builder
20             ->add('nomUser')
21             ->add('prenomUser')
22             ->add('email')
23             ->add('plainPassword', PasswordType::class, [
24                 'mapped' => false,
25                 'constraints' => [
26                     new NotBlank([
27                         'message' => "Veuillez entrer un mot de passe",
28                     ]),
29                     new Length([
30                         'min' => 6,
31                         'minMessage' => 'Votre mot de passe doit au minimum contenir {{ limit }} characters',
32                         'max' => 4096,
33                     ]),
34                 ],
35             ],
36         );
37     }
```

Utilisateur et sécurité

Afin de sécuriser mon projet j'ai ajouté deux rôles dans la partie **src/Entity/User.php** :

```
16     class User implements UserInterface
17     {
18         public const ROLE_USER = 'ROLE_USER';
19         public const ROLE_ADMIN = 'ROLE_ADMIN';
20     }
```

Ainsi qu'un nouveau « **construct** » qui affectera le rôle User à chaque nouvel utilisateur :

```
59         public function __construct()
60     {
61         $this->roles = [self::ROLE_USER];
62         $this->articles = new ArrayCollection();
63     }
```

Comme je l'expliquais plus haut mon rôle « USER » est en réalité les membres du bureau du club qui gère le site et le rôle « ADMIN » est en réalité l'équivalent d'un « Super Administrateur » c'est à dire le développeur, les besoins ne se prêtaient pas à davantage de rôles, il en sera peut-être le cas dans une futur mise à jour du site.

Afin d'affecter des autorisations à chaque utilisateur nous utiliserons l'annotation de methods **@IsGranted** à mettre en place dans les **controller** désiré ainsi que dans les **templates** afin d'afficher ou non certaines informations ou de limiter certains accès en fonction du rôle :

```
21  /**
22   * @Route("/", name="equipes_index", methods={"GET"})
23   * @IsGranted("ROLE_USER")
24   */
25  public function index(EquipesRepository $equipesRepository): Response
26  {
27      return $this->render('equipes/index.html.twig', [
28          'equipes' => $equipesRepository->findAll(),
29      ]);
30  }
```

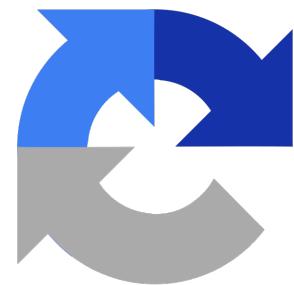
Dans cet exemple l'annotation `isGranted` signifie que seul les utilisateurs possédant le rôle `USER` et au dessus peuvent accéder à la page Index de l'entité `Equipes`.

Ensuite voici un exemple de l'utilisation de l'annotation **Twig is_granted** dans un fichier `html.twig` qui nous indique que si nous voulons accéder ou même voir le lien d'ajout d'un nouvel historique alors il faut être connecter et avoir le rôle « ADMIN » afin de pouvoir le voir et l'utiliser :

```
36  {% if is_granted("ROLE_ADMIN") %}
37  <div class="d-flex justify-content-end">
38  |   <a class="btnAdmin" href="{{ path('historique_new') }}>Ajouter un Historique</a>
39  |   </div>
40  {% endif %}
```

SÉCURITÉ RECAPTCHA

Il me fallait ensuite une solution pour éviter un maximum de robots spam sur mon site, j'ai donc installé un outil nommé **ReCAPTCHA**. Il s'agit d'un script que l'on appelle sur toutes les pages du site afin de les protéger d'éventuelles attaques.



reCAPTCHA

Veille sur la sécurité

Je souhaitais avoir une sécurité optimale pour mon site web, ce qui m'a poussé à faire de la veille en sécurité, on m'a conseillé de prendre en compte les actualités sur le site web **OWSAP** qui répertorie toutes les failles de sécurités web ainsi que les recommandations pour s'en prémunir.

www.owasp.org/www-project-top-ten/ ==> Liste des 10 types de risque liés à la sécurité d'application.

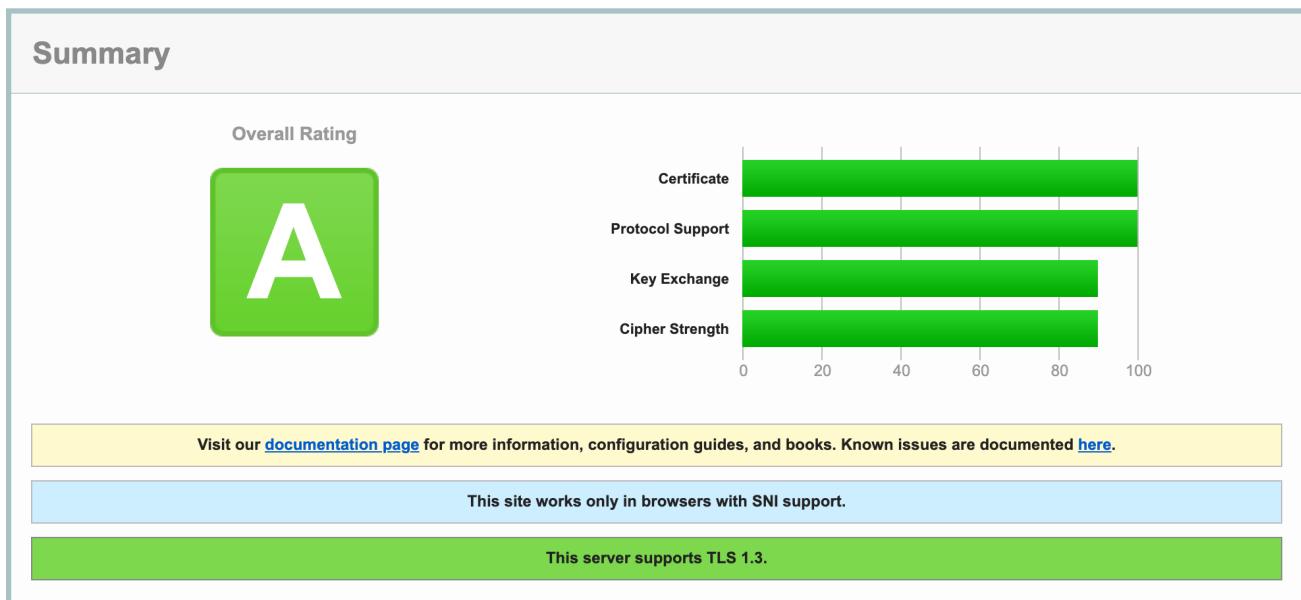
J'ai notamment pris le temps de lire la documentation Symfony, très bien expliquée et très complète ==> www.symfony.com/doc/current/security.html

Il y a aussi un site très intéressant pour les francophones c'est le site asfy.fr (Association Francophone Des Utilisateurs de Symfony) qui nous donne des petites astuces afin de mieux sécuriser nos sites web et notamment d'autre conseil pour d'autres domaines.

Un autre site qui mérite une petite mention c'est wanadev pour son article très intéressant, sur 5 sources d'amélioration pour la sécurité de votre application Symfony ==> [https://www.wanadev.fr/78-5-sources-d-amelioration-pour-la-securite-de-votre-application-symfony/](http://www.wanadev.fr/78-5-sources-d-amelioration-pour-la-securite-de-votre-application-symfony/)

SÉCURITÉ EN PRODUCTION

À partir du moment où mon site était en production et passée en **HTTPS** j'ai effectué d'autres tests de sécurité, premièrement avec le site Qualys où j'obtient un très bon score :



Detailed Blacklist Analyze

Global Blacklists	SPAM Blacklists
<ul style="list-style-type: none"> ● OK CMC Threat Intelligence ● OK CLEAN MX ● OK DNS8 ● OK MalSilo ● OK NotMining ● OK AICC (MONITORAPP) ● OK VX Vault ● OK securolytics ● OK Tencent ● OK MalwarePatrol ● OK Armis ● OK MalBeacon ● OK Comodo Valkyrie Verdict ● OK PhishLabs ● OK EmergingThreats ● OK Sangfor ● OK K7AntiVirus ● OK Virusdie External Site Scan ● OK Artists Against 419 ● OK CINS Army ● OK Cyren ● OK Cuttera ● OK AegisLab WebGuard ● OK MalwareDomainList ● OK Lumu ● OK zvelo ● OK Google Safebrowsing ● OK Kaspersky ● OK BitDefender ● OK GreenSnow ● OK iSum ● OK G-Data ● OK CyberCrime ● OK Malware Domain Blacklist ● OK AutoShun ● OK Feodo Tracker ● OK Web Security Guard ● OK Cyan ● OK CyRadar ● OK desennascara.me ● OK ADMINUSLabs ● OK SCUMWARE.org ● OK Dr.Web ● OK AlienVault ● OK Emsisoft ● OK Spamhaus ● OK malwares.com URL checker ● OK Phishrank ● OK EonScope ● OK MalwareD ● OK Avira ● OK Cisco Talos IP Blacklist ● OK OpenPhish ● OK Anti-AVL ● OK Forcepoint ThreatSeeker ● OK Spam404 ● OK Trustwave ● OK Certego ● OK URLhaus ● OK Yandex Safebrowsing ● OK ESET ● OK Threatsourcing ● OK Blocklist ● OK SecureBrain ● OK Nucleon ● OK PREBYTES ● OK Sophos ● OK Blueliv ● OK Hoplite Industries ● OK Netcraft ● OK CRDF ● OK ThreatHive ● OK BADWARE.INFO ● OK FraudScore ● OK Quick Heal ● OK Rising ● OK StopBadware ● OK Sucuri SiteCheck ● OK Fortinet ● OK StopForumSpam ● OK ZeroCERT ● OK Baidu-International ● OK Phishing Database ● OK McAfee ● OK Yandex ● OK Google 	<ul style="list-style-type: none"> ● OK ivmURI ● OK SEM FRESH ● OK SEM URI ● OK SEM UNRIED ● OK SORBS RHISBL BADCONF ● OK SORBS RHISBL NOMAIL ● OK Spamhaus DBL ● OK SURBL multi ● OK OSPAM ● OK Abuse.ro ● OK Abusix Mail Intelligence Blacklist ● OK Abusix Mail Intelligence Domain Blacklist ● OK Abusix Mail Intelligence Exploit list ● OK Anomnails DNSBL ● OK BACKSCATTERER ● OK BARRACUDA ● OK BLOCKLIST.DE ● OK CALIVENT ● OK CASA CBL ● OK CBL ● OK CYMRU BOGONS ● OK DAN TOR ● OK DAN TOREXIT ● OK DNS SERVICIOS ● OK DRMX ● OK DRONE BL ● OK FABELSOURCES ● OK HIL ● OK HIL2 ● OK Hostkarma Black ● OK IBM DNS Blacklist ● OK ICMFORBIDDEN ● OK IMP SPAM ● OK IMP WORM ● OK INTERSERVER ● OK ivmSIP ● OK ivmSIP24 ● OK JIPPG ● OK KEMPTBL ● OK KISA ● OK Konstant ● OK LASHBACK ● OK LNSGBLOCK ● OK LNSGBULK ● OK LNSGMULTI ● OK LNSGOR ● OK LNSGSRC ● OK MADAVI ● OK MailBlacklist ● OK MAILSPIKE BL ● OK MAILSPIKE Z ● OK MSRBL Phishing ● OK MSRBL Spam ● OK NETHERRELAYS ● OK NETHERUNSURE ● OK NIXSPAM ● OK Nordspam BL ● OK Nordspam DBL ● OK NoSolicitado ● OK ORVEDB ● OK PSBL ● OK RATS Dyna ● OK RATS NoPtr ● OK RATS Spam ● OK RBLJP ● OK RSBBL ● OK SCHULTE ● OK SEM BACKSCATTER ● OK SEM BLACK ● OK Sender Score Reputation Network ● OK SERVICESNET ● OK SORBS BLOCK ● OK SORBS DUHL ● OK SORBS HTTP ● OK SORBS MISC ● OK SORBS NEW ● OK SORBS SMTP ● OK SORBS SOCKS ● OK SORBS SPAM ● OK SORBS WEB ● OK SORBS ZOMBIE ● OK SPAMCOP ● OK Spamhaus ZEN ● OK SPIBL DNSBL ● OK Suomispam Reputation ● OK SWING ● OK TRIUMF ● OK TRUNCATE ● OK UCEPROTECTL1 ● OK UCEPROTECTL2 ● OK UCEPROTECTL3 ● OK Wordys SMTP Blacklist ● OK WPBL ● OK ZapBL

J'ai ensuite fait un autre test qui vérifie la totalité des fichiers présents sur le site et détecte les malwares ainsi que les failles de sécurité éventuelles, il s'agit du site **SITEGUARDING**. Il effectue aussi une vérification des liens présent sur le site en comparants ces derniers avec une liste de lien blackisé.

Comme nous pouvons le constater aucun problème ici.

TEST UNITAIRE DE SECURITÉ

Il me fallait à présent réaliser des tests un peu plus ciblé sur mon projet, j'ai donc effectué des tests unitaires à l'aide du Framework Open Source **PHPUnit**.

Pour l'installer il m'a fallut utiliser le gestionnaire de package Composer, via mon terminal à la racine de mon projet avec ces deux commandes :

```
➔ pchb composer require --dev symfony/phpunit-bridge
```

```
➔ pchb composer require --dev symfony/browser-kit symfony/css-selector
```

En premier lieu je vais d'abord tester le bon fonctionnement de mes controllers en vérifiant que toutes mes routes mènent bien à une page.

Pour cela, dans le dossier « Test », je crée un dossier « Controllers » puis un fichier « nomEntitéControllerTest » en réalité le nom importe peu il faut qu'il soit logique en fonction du controller à tester.

Voici un exemple de vérification de controller :

```
1 <?php
2 namespace App\Tests\Controller;
3
4 use Symfony\Bundle\FrameworkBundle\Test\WebTestCase;
5
6 class HomeControllerTest extends WebTestCase
7 {
8     /** @test */
9     public function TestHomeController()
10    {
11        $client = static::createClient();
12        $client->request('GET', '/');
13        $this->assertEquals(200, $client->getResponse()->getStatusCode());
14    }
15 }
```

Ici en l'occurrence je vérifie que ma page « home » dont le lien est la route '/' existe.

Une fois mon fichier paramétré, je dois lancer le test. Pour cela j'entre la ligne de commande suivante :

```
➔ pchb php bin/phpunit
```

Je voudrais ensuite tester les formulaires, les formulaires sont souvent primordiaux dans le bon fonctionnement des sites. C'est pour cela qu'il est très important de veiller à ce que ceux-ci

fonctionnent correctement et qu'ils ne présentent pas de failles de sécurité. Ils faut donc les tester.

Pour ce faire, dans le dossier « Test », j'ai créé un dossier « Forms » dans lequel j'ai créé un fichier « nomEntitéFormTest ». Là aussi le nom importe peu mais il faut qu'il soit logique en fonction du formulaire que nous voulons tester :

```
1  <?php
2
3  namespace App\Tests\Form;
4
5  use App\Entity\Contacts;
6  use Symfony\Bundle\FrameworkBundle\Test\KernelTestCase;
7
8  class ContactsFormTest extends KernelTestCase{
9
10     /** @test */
11     public function testContactsForm(){
12
13         $contact=(new Contacts())
14             ->setNomContact('Bonjour')
15             ->setPrenomContact('Quentin')
16             ->setEmailContact('q.andreani@gmail.com')
17             ->setObjetContact('Bonjour aurevoir')
18             ->setMessageContact('AuRevoir');
19
20         self::bootKernel();
21         $error = self::$container->get('validator')->validate($contact);
22         $this->assertCount(0,$error);
23
24     }
25 }
```

Je relance ma commande php bin/phpunit :

```
PHPUnit 8.5.14 by Sebastian Bergmann and contributors.

Testing Project Test Suite
....
4 / 4 (100%)

Time: 589 ms, Memory: 32.00 MB

OK (4 tests, 4 assertions)
```

Tout s'est bien passée en l'occurrence j'ai fait exprès de mettre de bonne information mais j'ai testé auparavant avec des informations erroné et j'ai ensuite ajouté des asserts.

EXEMPLE DE CODE

Le gestionnaire d'image

Pour gérer les images de mon carrousel de la page d'accueil, je voulais une solution simple et efficace. Le but étant que l'utilisateur puisse ajouter et supprimer autant d'image qu'il souhaite, j'ai donc choisie une méthode utilisée dans un tutoriel.

Je vais vous décrire ici ma démarche.

LA BASE DE DONNÉES

Pour commencer il me fallait au moins deux entités, une qui contiendra les images que j'ai nommé, de manière très original, « Images » et une autre qui sera liée et utilisera les images, en l'occurrence j'ai créé une entité « Accueil » simplement pour les images d'accueil.

J'ai donc une entité « Accueil » et une entité « Images » liées par une relation « un à plusieurs » (One To Many) :

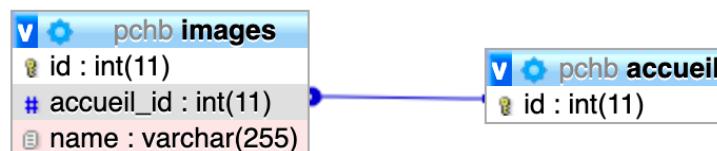


Image tiré, du mode « concepteur » de phpMyAdmin.

Je ne reviendrais pas sur les détails de la manière dont j'ai conçu les entités, simplement grâce à la commande `php bin/console make:entity`, ainsi que les commandes de migrations mentionnées plus haut.

LE CONTRÔLEUR

Une fois la base de donnée gérée, il me faut générer et configurer le controller de mon entité qui utilisera les images, en l'occurrence, l'entité « Accueil ». Pour ce faire j'ai taper la commande :

```
➔ pchb php bin/console make:crud Accueil
```

Cette commande m'a permis de générer automatiquement un controller ainsi que les formulaires nécessaire à l'ajout d'information dans mon entité.

Le formulaire :

Dans les formulaires d'ajout et de modification, la commande « make:crud » ne met pas en place la gestion des images, je vais donc devoir modifier le formulaire créé par défaut pour y ajouter les images.

Le formulaire d'ajout « d'accueil » est le fichier « src/Form/AccueilType.php », je modifie donc comme suit (des annotations sont ajoutées dans mon code) :

```
1  <?php
2
3  namespace App\Form;
4
5  use App\Entity\Accueil;
6  use Symfony\Component\Form\AbstractType;
7  use Symfony\Component\Form\FormBuilderInterface;
8  use Symfony\Component\OptionsResolver\OptionsResolver;
9  use Symfony\Component\Form\Extension\Core\Type\FileType;
10
11 class AccueilType extends AbstractType
12 {
13     public function buildForm(FormBuilderInterface $builder, array $options)
14     {
15         $builder
16             // J'ajoute le champ "images" dans le formulaire.
17             // Il ne faut pas le lié à la base de données ( mapped à false ).
18             ->add('images', FileType::class, [
19                 'label' => false,
20                 'multiple' => true,
21                 'mapped' => false,
22                 'required' => false
23             ])
24     ;
25 }
26
27 public function configureOptions(OptionsResolver $resolver)
28 {
29     $resolver->setDefaults([
30         'data_class' => Accueil::class,
31     ]);
32 }
33 }
```

Les méthodes du contrôleur :

Dans le contrôleur, je vais modifier plusieurs choses :

- ▶ Gestion des images ajoutées lors de la création d'un Accueil
- ▶ Gestion des images ajoutées lors de la modification d'un Accueil
- ▶ Suppression des images depuis le formulaire de modification

Avant toute chose je vais devoir copier les images dans un dossier physique situé dans le dossier « public ». Je vais modifier le fichier « service.yaml » et la section « parameters » en y inscrivant le code ci-dessous :

```
6     parameters:  
7         images_directory: '%kernel.project_dir%/public/uploads/images'
```

Ensuite dans les méthodes « new » et « edit » du contrôleur, je vais récupérer les données d'images, boucler dessus (ajout multiple possible) et les enregistrer sur le serveur et en base de données :

```
46         if ($form->isSubmitted() && $form->isValid()) {  
47             // Je récupère les images transmises  
48             $images = $form->get('images')->getData();  
49             // Je boucle sur les images  
50             foreach($images as $image){  
51                 // Je génère un nouveau nom de fichier  
52                 $fichier = md5(uniqid()).'.'.$image->guessExtension();  
53                 // Je copie le fichier dans le dossier uploads  
54                 $image->move(  
55                     $this->getParameter('images_directory'),  
56                     $fichier  
57                 );  
58                 // Je crée l'image dans la base de données  
59                 $img = new Images();  
60                 $img->setName($fichier);  
61                 $accueil->addImage($img);  
62             }  
63  
64             $entityManager = $this->getDoctrine()->getManager();  
65             $entityManager->persist($accueil);  
66             $entityManager->flush();  
67  
68             return $this->redirectToRoute('accueil_index');  
69         }
```

On ajoute le même code dans la partie « edit » du contrôleur.

À ce stade, je peux ajouter des images lors de la création et la modification d'accueil mais je ne peux pas les supprimer, je vais donc créer une méthode « **deleteImage** » qui nous permettra de supprimer une image.

Cette méthode sera appelée en Ajax en utilisant la méthode « **DELETE** » comme suit :

```

140     /**
141      * @Route("/supprime/image/{id}", name="accueil_delete_image", methods={"DELETE"})
142      * @isGranted("ROLE_USER")
143     */
144    public function deleteImage(Images $image, Request $request){
145        $data = json_decode($request->getContent(), true);
146
147        // Je vérifie si le token est valide
148        if($this->isCsrfTokenValid('delete'.$image->getId(), $data['_token'])){
149            // Je récupère le nom de l'image
150            $nom = $image->getName();
151            // Je supprime le fichier
152            unlink($this->getParameter('images_directory').'/'.$nom);
153
154            // On supprime l'entrée de la base
155            $em = $this->getDoctrine()->getManager();
156            $em->remove($image);
157            $em->flush();
158
159            // Je répond en json
160            return new JsonResponse(['success' => 1]);
161        }else{
162            return new JsonResponse(['error' => 'Token Invalide'], 400);
163        }
164    }
165 }
```

LES FICHIERS TWIG

Les fichiers Twig qui gèrent les formulaires devront être légèrement modifiés pour ajouter la gestion des images, principalement lors de la modification des annonces.

Je vais donc modifier les fichiers « **_form.html.twig** » et « **edit.html.twig** » du dossier « templates/accueil ». Dans le fichier « **_form.html.twig** » je vais afficher les images existantes si je suis sur la route « **accueil_edit** » :

```

1  {{ form_start(form) }}
2  {{ form_widget(form) }}
3  <div class="my-5"></div>
4  {% Si la route est "accueil_edit" on affiche les images %}
5  {% if app.request.attributes.get('_route') == 'accueil_edit' %}
6      <div class="d-flex">
7          {% for image in accueil.images %}
8              <div class="mx-3">
9                  
10                 <div class="text-center mt-4">
11                     {% J'ajoute un lien permettant de supprimer une image ( sera géré en Ajax ) %}
12                     <a class="btnAdmin" href="{{ path('accueil_delete_image', {id: image.id}) }}" data-delete data-token="{{ csrf_token('delete' ~ image.id) }}>Supprimer</a>
13                 </div>
14             </div>
15         {% endfor %}
16     </div>
17     {% endif %}
18     <div class="d-flex justify-content-end valide mt-5">
19         <button class="btnCustom"><span></span>{{ button_label|default('Ajouter') }}</button>
20     </div>
21  {{ form_end(form) }}
```

Dans le même fichier en dessous de la balise {{ form_end(form) }} je vais ajouter une balise « script » qui chargera le fichier « image.js » qui sera créé à l'étape suivante :

```
23  {% block javascripts %}  
24      <script src="{{ asset('js/images.js') }}"></script>  
25  {% endblock %}
```

LE JAVASCRIPT

Je vais à présent ajouter du javascript permettant de gérer la suppression des images lors du clic sur le lien « Supprimer » à côté des images. Le fichier est le suivant :

```
1  window.onload = () => {  
2      // Gestion des boutons "Supprimer"  
3      let links = document.querySelectorAll("[data-delete]")  
4  
5      // Je boucle sur links  
6      for(link of links){  
7          // J'écoute le clic  
8          link.addEventListener("click", function(e){  
9              // J'empêche la navigation  
10             e.preventDefault()  
11  
12             // Je demande confirmation  
13             if(confirm("Voulez-vous supprimer cette image ?")){  
14                 // J'envoie une requête Ajax vers le href du lien avec la méthode DELETE  
15                 fetch(this.getAttribute("href"), {  
16                     method: "DELETE",  
17                     headers: {  
18                         "X-Requested-With": "XMLHttpRequest",  
19                         "Content-Type": "application/json"  
20                     },  
21                     body: JSON.stringify({_token: this.dataset.token})  
22                 }).then(  
23                     // Je récupère la réponse en JSON  
24                     response => response.json()  
25                 ).then(data => {  
26                     if(data.success)  
27                         this.parentElement.remove()  
28                     else  
29                         alert(data.error)  
30                 }).catch(e => alert(e))  
31                 // Je rafraîchit la page  
32                 location.reload();  
33             }  
34         })  
35     }  
36 }
```

RÉALISER UNE INTERFACE UTILISATEUR AVEC GESTION DE CONTENU

Pour répondre au besoins du club sur la personnalisation du site, j'ai mis en place un système de gestion de contenu. Le club aura la possibilité d'importer ses photos, de mettre du texte ou de le modifier.

INTERFACE D'ADMINISTRATION

La partie Administration du site est relativement simple.

J'ai créé une nouvelle barre de navigation en deux partie, qui comprend tout le haut et la gauche de la page en position fixe.

La partie haute comprend le bouton d'accueil de la partie Administration, un bouton pour retourner au site et un bouton de déconnexion.

The screenshot shows a dark-themed administrative dashboard. At the top, there's a navigation bar with the word "ADMINISTRATION" in white. To its right are three links: "Accueil" (in red), "Retour au site", and "Déconnexion". On the far left, there's a vertical sidebar containing icons for various functions: a euro sign, a document, a user profile, a Facebook icon, a briefcase, an envelope, a clipboard, a gear, a grid, a circular arrow, a camera, and a folder. The main content area has a header "ACCUEIL ADMINISTRATION" with a red underline. Below it, a message says "Bonjour Quentin" followed by "Voici vos deux derniers messages". Two messages are listed:

Auteur :	Email :	Objet :	Contenu (extrait)	Action
MarcusTullius Cicero	marcustulliuscicero@spqr.rom	De finibus bonorum et malorum	Ed ut perspiciatis, unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam eaque ipsa, quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt, explicabo. Nemo enim ipsam voluptatem, quia voluptas sit, aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos, qui ratione voluptatem sequi nesciunt, neque porro quisquam est, qui dolorem ipsum, quia dolor sit, amet, consecetur, adipisci velit, sed quia non numquam eius modi tempora incident, ut labore et dolore magnam aliquam quaerat voluptatem.	
Marc Aurel	marcaurel@spqr.rom	De finibus bonorum et malorum	Ed ut perspiciatis, unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam eaque ipsa, quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt, explicabo. Nemo enim ipsam voluptatem, quia voluptas sit, aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos, qui ratione voluptatem sequi nesciunt, neque porro quisquam est, qui dolorem ipsum, quia dolor sit, amet, consecetur, adipisci velit, sed quia non numquam eius modi tempora incident, ut labore et dolore magnam aliquam quaerat voluptatem.	

Page d'accueil de la partie « Administration » du site.

La partie gauche de la barre de navigation comprend tout les liens de navigation dans la partie Administration du site.

La bar de navigation s'agrandie au « hover » lorsque la souris passe par dessus et laisse apparaître le nom des liens de navigation couplet aux icônes pour plus d'ergonomie :

The screenshot shows the 'ADMINISTRATION' page with a dark blue header containing the title 'ADMINISTRATION' and three links: 'Accueil', 'Retour au site', and 'Déconnexion'. Below the header is a sidebar with a white background and a dark blue border. The sidebar lists several navigation items with icons and descriptions:

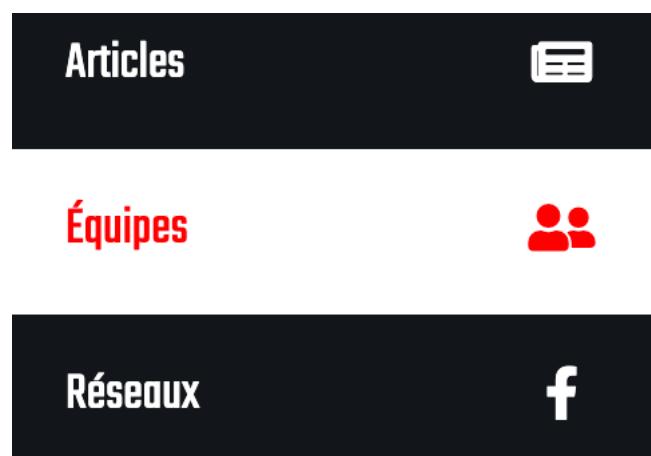
- Tarifs** (€) - Description: 'Articles' (document icon)
- Équipes** (User icon) - Description: 'Quentin' (red text)
- Réseaux** (Facebook icon) - Description: 'Quentin messages' (red text)
- Sponsors** (Briefcase icon) - Description: 'MarcusTullius Cicero' (text), 'tulliuscicero@spqr.rom' (text), 's bonorum et malorum' (text), and a long Latin text block.
- Messages** (Email icon) - Description: 'Marcus Aurel' (text), 'marcaurel@spqr.rom' (text), 's bonorum et malorum' (text), and a long Latin text block.
- Inscriptions** (Document icon)
- Communauté** (Globe icon)
- Organigramme** (Hierarchical icon) - Description: 'Marc Aurel' (text), 'marcaurel@spqr.rom' (text), 's bonorum et malorum' (text), and a long Latin text block.
- Historique Club** (Clock icon)
- Images Accueil** (Image icon)

At the bottom left of the sidebar, there is a link: <https://www.pchb.quentin-andreani.fr/public/tarifs/>

La bar de navigation lorsque la souris passe par dessus, un effet « current » à aussi été appliquée.

GESTION DES RUBRIQUES

Pour en revenir quelques instants sur la bar de navigation un lien ce présente comme ceci :



Si je clique sur un des liens (Pour l'exemple le lien « Équipes »), j'arrive sur une page d'index ou de modification de la rubrique sélectionnée. Dans mon exemple pour les équipes le liens nous amène dans la page d'index des équipes.

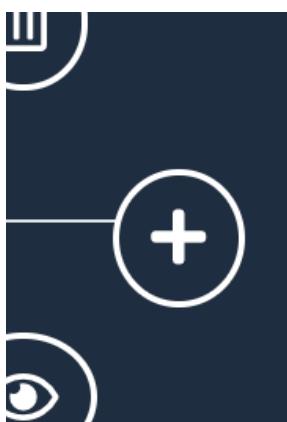
Cette dernière est représenté sous la forme d'un tableau avec des boutons pour interagir avec, ainsi l'utilisateur à la possibilité (dans cette exemple) de Voir la page lié à l'entité (représenté par l'icône « œil », de l'édition (icône « crayon »), ou bien de la supprimer (icône « poubelle »)).

ADMINISTRATION				
	LES ÉQUIPES			Retour
	Nom	Image	Contenu	Actions
	Senior Masculin		<p>Senior M1- Saison 2019-2020</p> <p>Lore ipsum dolor sit amet, consectetur adipiscing elit. Mauris quis vehicula metus. In vulputate ipsum in elit vestibulum tempor. Etiam pharetra quam sed neque euismod ultricies. Duis gravida, turpis et fringilla suscipit, orci nulla pulvinar urna, ac fermentum orci orci non quam. Donec lacinia ipsum ut nunc scelerisque rhoncus. Praesent volutpat nunc vel leo volutpat, vel tempor quam tempus. Nam vel quam sed ante dictum consequat vitae vitae quam. Ut ulvinar dui leo. Etiam a justo ac enim varius porttitor ut a ex. Curabitur in dolor sem.</p> <p>Classement et statistiques des séniors M1</p>	
	-19 Masculin		<p>Saison 2019-2020</p> <p>Les -19M sont coachés par Stéphane</p>	

Ici l'exemple est pour l'entité « Equipes » mais le concept est le même pour la plupart des autres rubriques.

AJOUT D'UNE RUBRIQUE

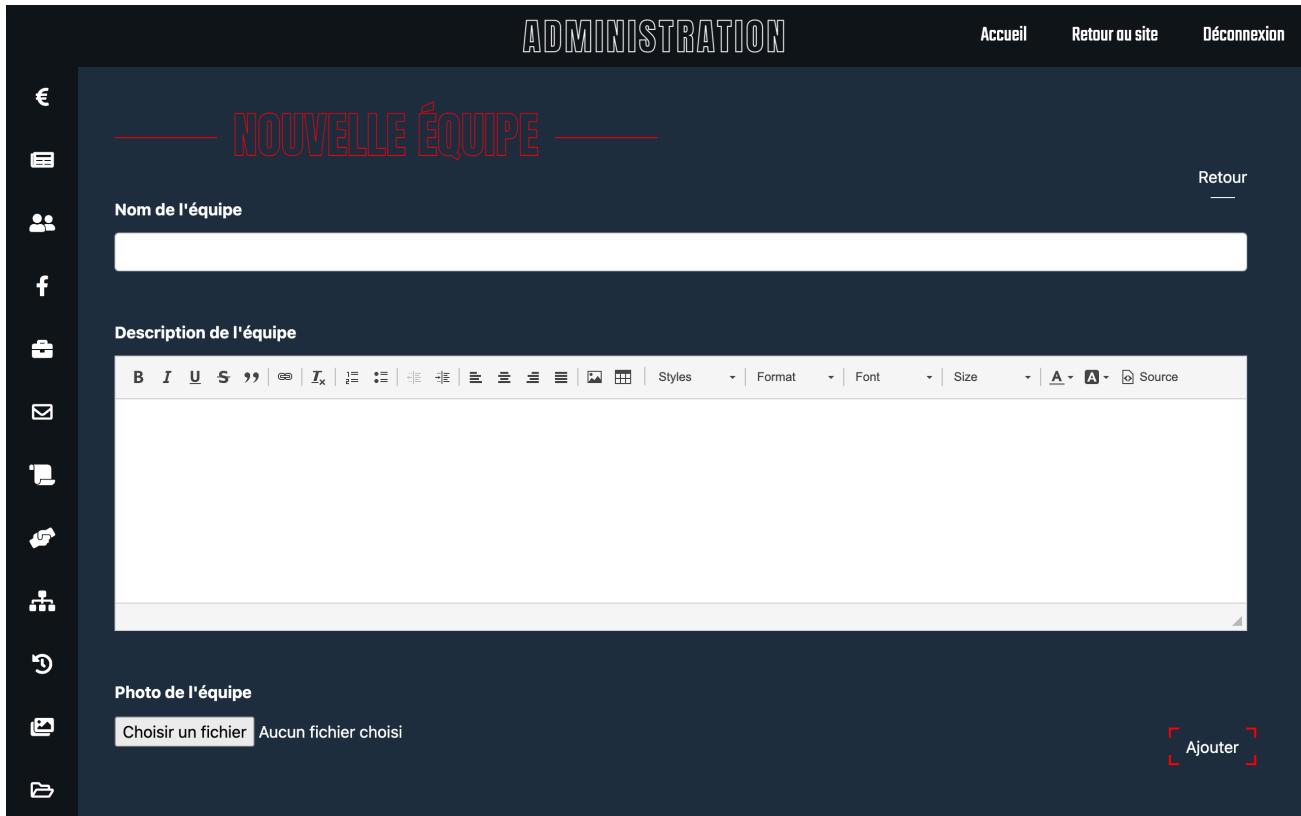
Afin d'ajouter une entité dans la rubrique, en l'occurrence une équipe pour l'exemple (globalement le concept est le même pour les autres parties de la gestion de contenu), j'ai mis en place un boutons « plus » en bas à droite de l'écran en position « fixed » à la manière des applications google afin qu'il reste dans le champ de vision de l'utilisateur.



Quand on passe la souris par dessus, les différents boutons changent de couleur.



En cliquant sur le bouton « + » l'utilisateur est dirigé sur la page d'ajout de contenu, en l'occurrence « Nouvelle Équipe » qui se présente comme suit :



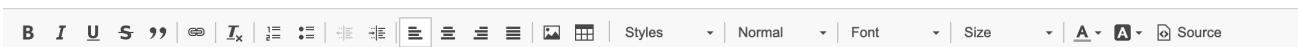
Ici il y a trois champs à remplir le premier étant le nom de l'équipe, le second est le contenu, la description de l'équipe est le troisième est l'image de l'équipe.

Pour ce qui est du système de traitement de texte du contenu j'ai utilisé **CK Editor**, qui est un éditeur de texte open-source, que j'ai installé via **composer sous forme de bundle** dans symfony.

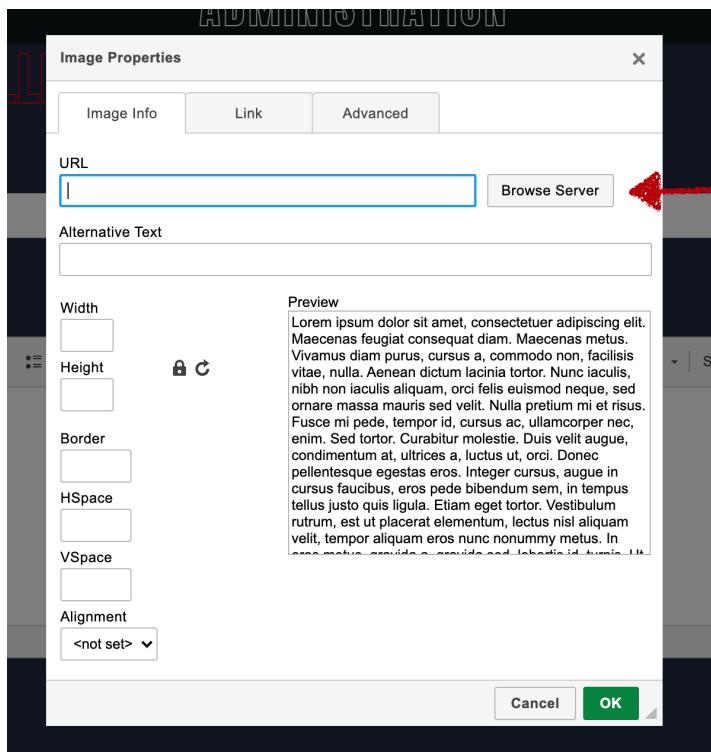
J'ai ensuite paramétré la bar d'outil dans le fichier **fos_ckeditor.yaml** comme suit :

```
fos_ck_editor:
    configs:
        main_config:
            filebrowserBrowseRoute: elfinder
            filebrowserBrowseRouteParameters: []
            toolbar:
                - { name: "styles", items: ['Bold', 'Italic', 'Underline', 'Strike', 'Blockquote', '-', 'Link', '-', 'RemoveFormat', '-', 'NumberedList', 'BulletedList', '-', 'Outdent', 'Indent', '-', '-', 'JustifyLeft', 'JustifyCenter', 'JustifyRight', 'JustifyBlock', '-', 'Image', 'Table', '-', 'Styles', 'Format', 'Font', 'FontSize', '-', 'TextColor', 'BColor', 'Source'] }
```

Il me fallait aussi une solution pour que les utilisateurs puissent ajouter des images dans CK Editor afin d'enrichir leurs descriptions ou leurs articles, ce pourquoi j'ai utilisé une solution intéressante d'upload et de gestion d'image qui est très utile avec CK Editor : **ElFinder**. Elfinder s'installe aussi comme un bundle via composer.



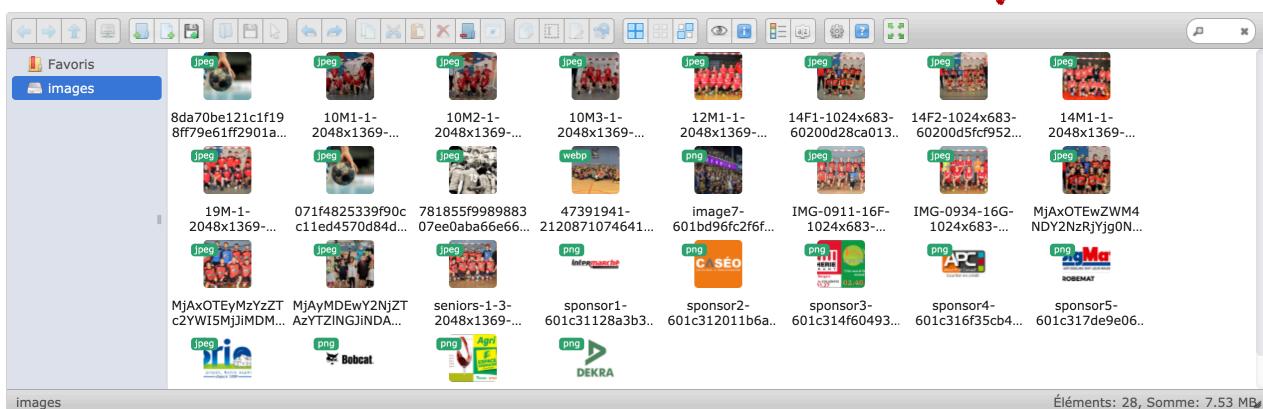
Pour y accéder il nous faut en premier lieu cliquer sur l'icône « image » de la bar d'outil



Voici la fenêtre qui apparaît lorsque nous cliquons sur l'icône image.

Le bouton « Browse Server » n'existe pas avant l'installation de Elfinder, ce bouton va nous permettre d'accéder à l'interface de Elfinder.

Nous arrivons ensuite sur l'interface de Elfinder



Elfinder est un plugin qui permet aux administrateurs de parcourir l'arborescence d'un site SPIP à travers un outil ressemblant à un gestionnaire de fichiers. Dans ce plugin, on s'en sert non pour gérer des fichiers et des dossiers, mais pour gérer des articles et des rubriques.

On retrouve les fonctionnalités basiques des gestionnaires de fichiers, c'est à dire déplacer, copier, créer un dossier, charger des fichiers. Pour ajouter l'image il suffit de double cliquer dessus, de régler sa taille et de valider avec OK.

Pour finir il suffit de cliquer sur le bouton « Ajouter » et le tour est joué.



EDITION / SUPPRESSION D'UNE RUBRIQUE

Pour éditer ou supprimer une rubrique ou en l'occurrence une équipe il nous suffit de retourner sur la page d'index des équipes et de cliquer sur le bouton désiré.

Je précise que la page d'édition est en tout point similaire à la page d'ajout d'une rubrique, il y a simplement un lien en plus pour supprimer dans la page d'édition.

Pour la suppression d'une rubrique, nous devons donc accéder à la page d'édition et cliquer sur le lien « Supprimer l'équipe » :

The screenshot shows the 'ADMINISTRATION' interface with the 'EDITER ÉQUIPE' (Edit Team) section active. On the left, a sidebar contains icons for various administrative tasks. The main area displays team details: 'Nom de l'équipe' (Senior Masculin), 'Description de l'équipe' (Senior M1- Saison 2019-2020 with a long Lorem ipsum placeholder), and 'Photo de l'équipe' (button to choose a file). A red arrow points from the bottom-left towards the 'Supprimer l'équipe' (Delete Team) button in the bottom right corner of the main form. This button is highlighted with a red border. A second red arrow points from the bottom of the main form down to a modal dialog box. The modal has a dark background and contains the text: 'www.pchb.quentin-andreani.fr indique' and 'Etes-vous sur de vouloir supprimer cette équipe ?' (Are you sure you want to delete this team?). It features 'Annuler' (Cancel) and 'OK' buttons.

Une modal s'affichera alors, demandant à l'utilisateur une confirmation pour la suppression la rubrique.

FRONT-END

Mise en place du Framework CSS

Symfony ne possède pas nativement Bootstrap mais est conçue pour fonctionner très efficacement avec, la mise en place est plutôt simple.

Pour utiliser Bootstrap avec le CDN il me faut poster dans mon <head> ces deux balises :

```
9 |     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-giJF6kkoqNQ00vy+HMDP7az0uL0xtbfIcaT9wjKHR8RbDVddVHyTfAAAsrekwKmP1" crossorigin="anonymous">
10|     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/js/bootstrap.bundle.min.js" integrity="sha384-ygbV9kiqUc6oa4msXn9868pttMgiQaeYH7/t7LECLbyPA2x65Kgf800JFdroafW" crossorigin="anonymous"></script>
```

Et en bas de page je dois appeler la bibliothèque JQuery :

```
105 |     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
```

À ce niveau là je peux déjà utiliser Bootstrap, cependant je voulais aussi utiliser Bootstrap nativement pour tout mes formulaires j'ai donc ajouter ce paramètre dans « **config/packages/twig.yaml** » :

```
1 ▼ twig:
2   |   default_path: '%kernel.project_dir%/templates'
3   |   form_themes: ['bootstrap_4_layout.html.twig']
```

Ce paramètre me permettra d'avoir nativement une mise en forme Bootstrap sur tout mes formulaires.

SASS ET JS

J'ai décidé pour le CSS t'utiliser SASS, en partie pour la gestion plus simple du code, son système **@extend**, et son système **@import**. Ma démarche est simple je crée un nouveau fichier **.scss** pour chaque grande partie de mon site et j'importe le tout dans un fichier « **style.scss** » que je transforme en suite via la commande :

```
➔ styles sass –watch style.scss:style.css
```

Le fichier **style.scss** comportant tout les **@import** de mes différent fichier se compile donc en fichier **style.css**.

Voici mon fichier style.scss

```
1  **** Fichier SCSS Main ****
2
3
4  /*
5   - Document: Fichier style d'appel
6   - Version: 1.0.0
7   - Author: Andreani Quentin
8 */
9 @import url('https://fonts.googleapis.com/css2?family=Anton&display=swap');
10 @import url('https://fonts.googleapis.com/css2?family=Teko:wght@300;400;500;600;700&display=swap');
11 @import url('https://fonts.googleapis.com/css2?family=Barlow:wght@200;300;400&display=swap');
12 @import "body.scss";
13 @import "admin.scss";
14 @import "header.scss";
15 @import "accueil.scss";
16 @import "register&login.scss";
17 @import "equipes.scss";
18 @import "actualites&articles.scss";
19 @import "contacts.scss";
20 @import "inscription.scss";
21 @import "footer.scss";
```

Voici un exemple de l'utilisation du système d'imbrication de SASS :

Je trouve pour ma part que le système d'imbrication de Sass particulièrement ingénieux et pratique, à présent, je l'utilise en permanence.

```
125 **** Page Administration ****
126
127 .admin {
128     h3 {
129         color: red;
130         font-weight: 300;
131     }
132     .ligne {
133         height: 1px;
134         border-bottom: 1px solid red;
135         width: 600px;
136
137         @media screen and (max-width: 600px) {
138             width: 400px;
139         }
140         @media screen and (max-width: 400px) {
141             width: 250px;
142         }
143     }
144     .messageAdmin {
145         color: white;
146         .message {
147             text-align: justify;
148         }
149     }
150     .options {
151         border-right: 1px solid red;
152     }
153     .userAdmin {
154         table {
155             color: white;
156         }
157     }
158 }
```

Réaliser une interface web statique et dynamique

Pour que l'expérience utilisateur soit la plus intuitive et interactive possible, il me fallait rendre mon interface responsive adaptable sur plusieurs tailles d'écrans, types d'appareils.

Je me suis servis, tout d'abord, des différentes class Bootstrap « **sm - md - lg - xl** » afin que les éléments s'adapte au mieux à l'écran en fonction du ratio et de l'orientation. Leur utilisation est plutôt simple, on part sur une grille de 12 colonnes qu'il nous faut remplir avec la class « **col** » et des lignes avec la class « **row** », le tout peut être contenu dans une div contenant la class « **container** ».

Exemple :

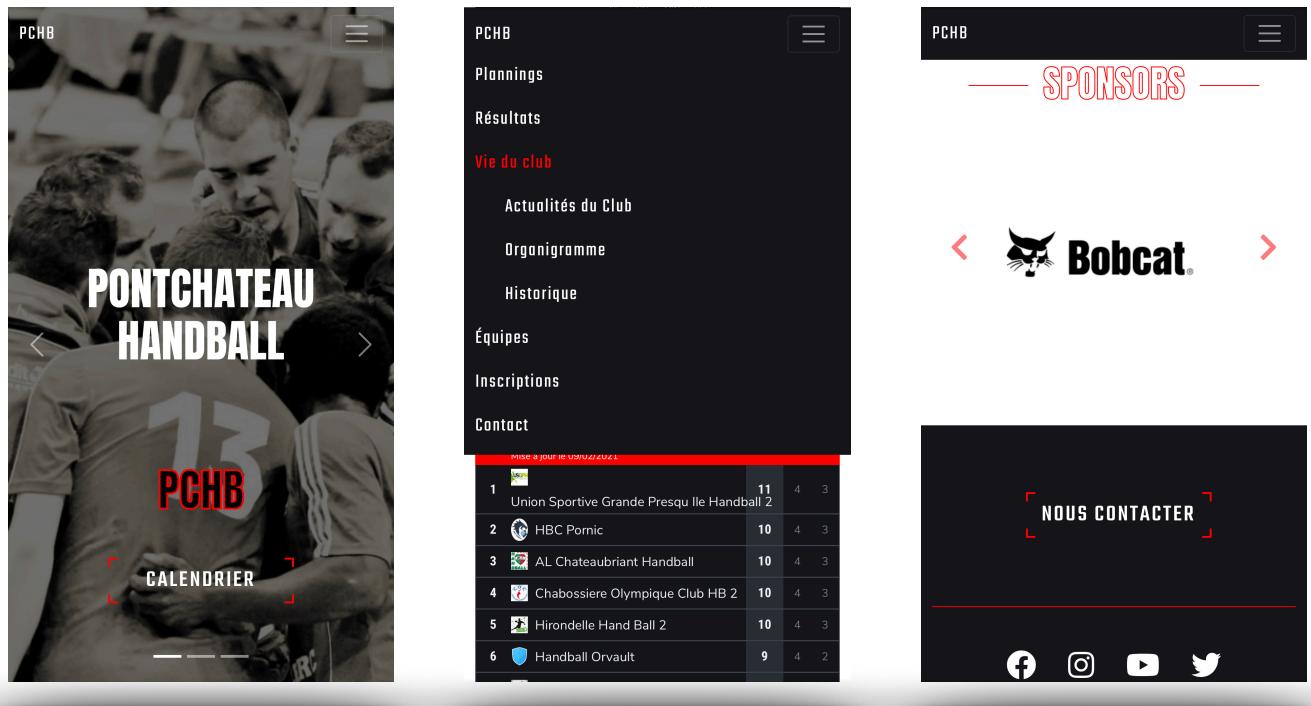
```
93     <section class="actualites">
94         <div class="container">
95             <h2>Actualités</h2>
96             {% for article in articles|reverse %}
97                 {% if loop.first %}
98                     <div class="row">
99                         <a href="{{ path('articles_show', {'id': article.id}) }}>
100                             <div class="col-lg-12 actualites_accueil_un" style="background-image: url('{{ asset('uploads/images/' ~ article.
101                             imageArticle) }});">
102                                 <div class="row child">
103                                     <div class="col-lg-8"></div>
104                                     <div class="col-lg-4">
105                                         <div class="titreArticle">
106                                             <p>{{ article.dateArticle|format_date('none', 'dd MMMM Y', null, 'gregorian', 'fr') }}</p>
107                                             <h4>{{ article.titreArticle }}</h4>
108                                         </div>
109                                     </div>
110                                 </div>
111                             </a>
112                         </div>
113                     {% endif %}
114                 {% endfor %}
```

Un autre exemple intéressant de l'utilisation de Bootstrap est la **navbar**, cette dernière est très facile à mettre en place mais il est plus compliqué de la personnaliser (idem pour le carousel). Le gros avantage de la navbar Bootstrap est qu'elle est responsive et qu'elle affiche donc naturellement un menu burger sur les petits écrans :



AFFICHAGE MOBILE

Pour l'affichage mobile j'ai dû utiliser malgré Bootstrap de nombreuse **media queries**, réduire la taille des icônes, de la font et des images. Voici un exemple de l'interface mobile :



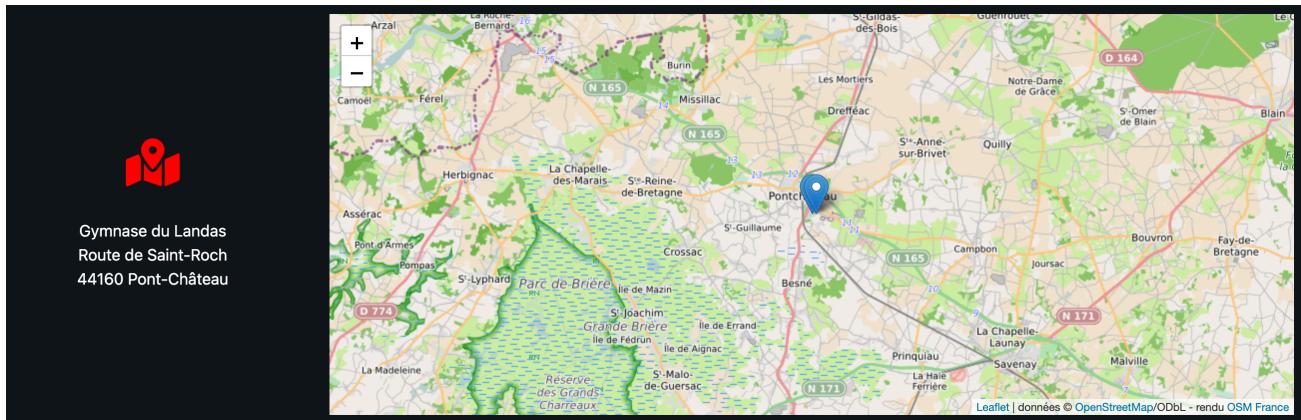
Exemple d'une media queries :

Voici une Media queries correspondant à l'affichage tablette et mobile afin de réduire la taille de la petite ligne de séparation entre mon bouton contact et les icônes réseaux dans mon footer.

```
68     /**** Icône Réseaux & Mentions ****/
69
70 ▼ .ligne {
71     height: 1px;
72     border-bottom: 1px solid red;
73     width: 600px;
74
75     @media screen and (max-width: 600px) {
76         width: 400px;
77     }
78     @media screen and (max-width: 400px) {
79         width: 250px;
80     }
81 }
82 }
```

API OPEN STREET MAP

Afin de rajouter un peu de dynamisme au site, j'ai utilisé la librairie javascript Leaflet couplé à Open Street Map qui est un projet de carte Open Source et gratuite. J'ai configuré le fichier et lui ai rajouté un marqueur afin de pouvoir identifier les lieux d'entrainements du club :



Pour situer la carte dans le cadre il me faut utiliser ce script :

```
30 |         <script type="text/javascript">
31 |
32 |             var lat = 47.42836769159643;
33 |             var lon = -2.082151696398331;
34 |             var macarte = null;
35 |
36 |             var salles = {
37 |                 "Landas": { "lat": 47.42836769159643, "lon": -2.082151696398331 },
38 |                 "Galfione": { "lat": 47.42875867917015, "lon": -2.0790177482454637 }
39 |             };
40 |             // Fonction d'initialisation de la carte
41 |             function initMap() {
42 |                 // Créer l'objet "macarte" et l'insérer dans l'élément HTML qui a l'ID "map"
43 |                 macarte = L.map('map').setView([lat, lon], 11);
44 |                 // Leaflet ne récupère pas les cartes (tiles) sur un serveur par défaut. Nous devons lui préciser où nous
45 |                 // souhaitons les récupérer. Ici, openstreetmap.fr
46 |                 L.tileLayer('https://s.tile.openstreetmap.fr/osmfr/{z}/{x}/{y}.png', {
47 |                     // Il est toujours bien de laisser le lien vers la source des données
48 |                     attribution: 'données ©<a href="//osm.org/copyright">OpenStreetMap</a>/ODbL - rendu <a href="//openstreetmap.fr">OSM France</a>',
49 |                     minZoom: 1,
50 |                     maxZoom: 20
51 |                 }).addTo(macarte);
52 |
53 |                 for (salle in salles) {
54 |                     var marker = L.marker([salles[salle].lat, salles[salle].lon]).addTo(macarte);
55 |                 }
56 |             }
57 |             window.onload = function(){
58 |                 // Fonction d'initialisation qui s'exécute lorsque le DOM est chargé
59 |                 initMap();
60 |             };
61 |         </script>
```

On déclare les variables lat et lon pour l'emplacement de la carte et on déclare ensuite les longitudes et latitudes dans un tableau des différents endroits que nous voulons marquer.

GOOGLE ANALYTICS

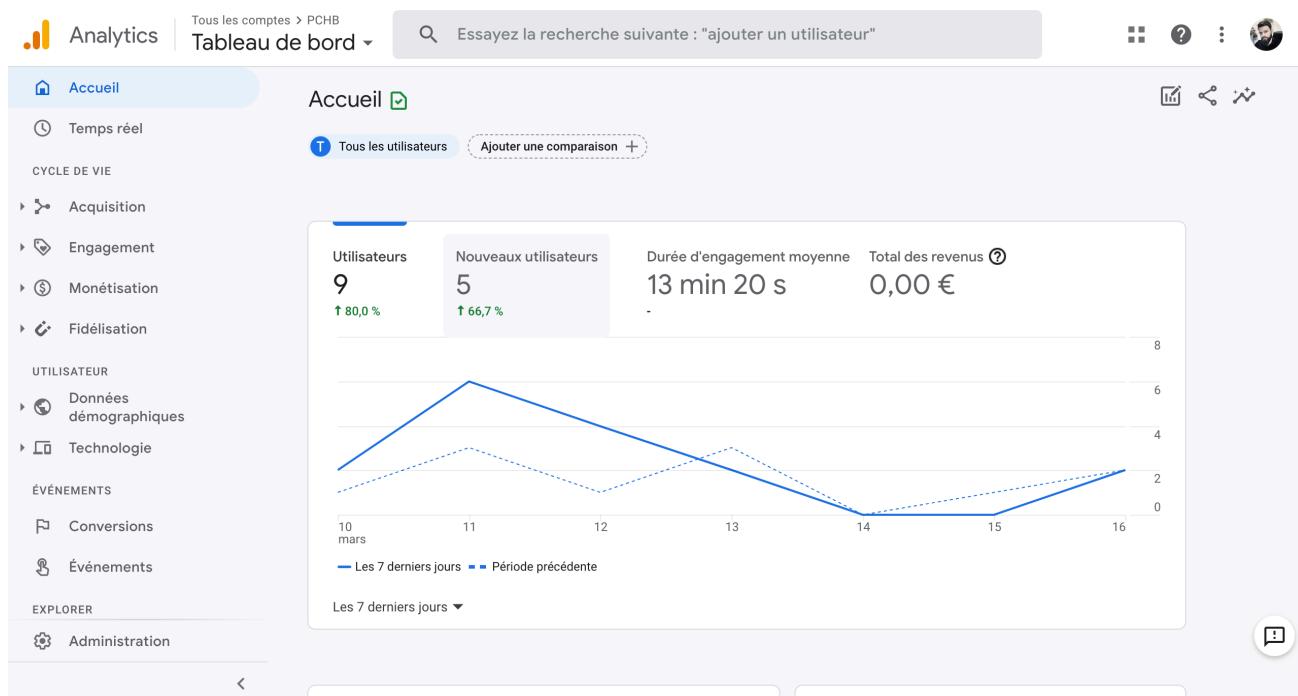
Afin d'analyser le trafic sur le site, j'ai installé le plugin Google Analytics pour avoir un récapitulatif des taux d'affluence.

Voici le code que j'ai du intégrer pour faire fonctionner Google Analytics :

```
<!-- Global site tag (gtag.js) - Google Analytics -->
<script async src="https://www.googletagmanager.com/gtag/js?id=G-8NVPYW9KG7"></script>
```

Google récupère les informations du site grâce à ce « tag ».

Et voici mon tableau de bord pour le site pchb :



Je l'utilise aussi pour mon portfolio sur quentin-andreani.fr

Il faut savoir avant de parler de la mise en production, que j'ai utilisé mon nom de domaine avec un sous-domaine pour mettre mon site en ligne, et cela simplement pour réaliser des tests, le site sera migré sur l'hébergement du club qui est un hébergement OVH.

MISE EN PRODUCTION

Pour ce qui est de la mise en production j'ai longuement hésité entre OVH et o2switch, mon choix s'est finalement porté sur O2switch, son offre me paraissait plus intéressante et plus complète qu'OVH.

O2switch est un entreprise 100% Française autant pour le serveur que pour le SAV.



Une phase importante du développement est

le recettage, c'est ce qui permet d'engager une multitude de test que ce soit au niveau du sport (Mobile, Tablette) mais aussi faire des tests de sécurité, tout ceci permettant de vérifier si son site ne comporte pas de failles (j'en parle un peu plus dans la partie sécurité).

Pour cette mise en production il me fallait un logiciel pour Mac OS qui me permette de me connecter au serveur FTP de l'hébergement o2switch, pour ce faire j'ai trouvé le logiciel **Cyberduck** qui est un logiciel client FTP et SFTP pour la plateforme Mac OS X.



Cyberduck est un logiciel très facile d'utilisation avec une interface moderne et efficace.

Je m'en suis servi pour envoyer l'intégralité de mon projet Symfony, dans le dossier de mon sous-domaine pchb.quentin.andreani.fr, mis à part le dossier **var** et **vendor** que je trouvais trop volumineux et que je pouvais rétablir via la ligne de commande « **composer update** » via le terminal (disponible sur l'interface **Cpanel**) mis à disposition par o2switch une fois mes fichiers dans le serveur.

Il me fallait ensuite créer un fichier **.htaccess** et le placer dans le dossier « **public** » depuis la racine de mon dossier Symfony. Le fichier **.htaccess** est un fichier de configuration (Apache) permettant de définir des règles bien spécifiques dans un répertoire. Ce type de fichier peut être utilisé pour réaliser des redirections ou protéger un répertoire par un mot de passe.

Il me fallait également un fichier **.env** à la racine de mon projet afin de configurer l'accès à la base de donnée.

The screenshot shows the o2switch CPanel interface. At the top, there's a navigation bar with the o2switch logo, a search bar, and user information (nlkj9922). Below the navigation is a sidebar with various icons and links:

- RECOMMANDATIONS**: Includes links to WPChef, WP Marmite, WP Rocket, SecuPress Pro, SEOPress Pro, and WP Assistance.
- DOMAINES**: Includes links to Domaines Configurés, Sous-domaines, Redirections, and Zone Editor.
- E-MAIL**: Includes links to Comptes de messagerie, Redirecteurs, Routage des e-mails, Réponses automatiques, Listes de diffusion, and Suivre les messages sortants.
- System Status** on the right side:
 - Utilisation Du Disque PostgreSQL: 0 octet / ∞
 - Bande Passante: 535,02 MB / ∞
 - Domaines Configurés: 1 / ∞
 - Sous-Domains: 5 / ∞
 - Comptes De Messagerie: 1 / ∞
 - Listes De Diffusion: 0 / ∞
 - Réponses Automatiques: 0 / ∞
 - Redirecteurs: 0 / ∞
 - Filtres De Messagerie: 0 / ∞
 - Comptes FTP: 0 / ∞

A URL bar at the bottom shows: https://buffle.o2switch.net:2083/cpsess7139016544/frontend/paper_lantern/elegantschool/index.html

L'interface CPanel d'o2Switch, très complète

Étant plutôt à l'aise avec les lignes de commande **Bash**, j'utilise beaucoup le terminal de l'interface CPanel pour régler mes problèmes déplacer mes fichiers, en supprimer ou même parfois en éditer avec **VIM**.

The screenshot shows the o2switch CPanel Terminal interface. The title bar says "Terminal". The main area displays a command-line session:

```
[nlkj9922@buffle ~]$ cd pchb.quentin-andreani.fr
[nlkj9922@buffle pchb.quentin-andreani.fr]$ ls -la
total 444
drwxr-x--- 15 nlkj9922 nobody 4096 Mar 11 22:13 .
drwxr-x--- 26 nlkj9922 nlkj9922 4096 Mar 11 21:05 ..
-rw-r---- 1 nlkj9922 nlkj9922 1419 Feb 7 18:02 .env
-rw-r---- 1 nlkj9922 nlkj9922 162 Jan 26 10:51 .env.test
-rw-r---- 1 nlkj9922 nlkj9922 298 Jan 26 10:51 .gitignore
-rw-r---- 1 nlkj9922 nlkj9922 157 Mar 11 22:13 .htaccess
drwxr-xr-x 2 nlkj9922 nlkj9922 4096 Feb 7 18:25 idea
drwxr-xr-x 3 nlkj9922 nlkj9922 4096 Feb 8 12:24 .well-known
-rw-r---- 1 nlkj9922 nlkj9922 229 Feb 7 18:03 400.shtml
-rw-r---- 1 nlkj9922 nlkj9922 207 Feb 7 18:03 401.shtml
-rw-r---- 1 nlkj9922 nlkj9922 203 Feb 7 18:03 403.shtml
-rw-r---- 1 nlkj9922 nlkj9922 203 Feb 7 18:03 404.shtml
-rw-r---- 1 nlkj9922 nlkj9922 216 Feb 7 18:03 413.shtml
-rw-r---- 1 nlkj9922 nlkj9922 243 Feb 7 18:03 500.shtml
drwxr-xr-x 2 nlkj9922 nlkj9922 4096 Feb 7 18:25 bin
drwxr-xr-x 2 nlkj9922 nlkj9922 4096 Feb 7 18:03 cgi-bin
-rw-r---- 1 nlkj9922 nlkj9922 3069 Feb 8 21:59 composer.json
-rw-r---- 1 nlkj9922 nlkj9922 312689 Feb 8 22:13 composer.lock
drwxr-xr-x 4 nlkj9922 nlkj9922 4096 Mar 11 19:54 config
-rw-r---- 1 nlkj9922 nlkj9922 10404 Feb 7 18:03 cp_errordocument.shtml
drwxr-xr-x 2 nlkj9922 nlkj9922 4096 Feb 7 18:26 migrations
```

At the bottom, there are links to Accueil, Marques déposées, Privacy Policy, Documentation, and Aide.

Le terminal CPanel d'O2Switch très pratique pour gérer de nombreuses choses, et surtout lancer un nettoyage du cache à la racine du projet avec un « **php bin console clear:cache** »

LE RÉFÉRENCEMENT

Afin d'avoir un bon référencement sur le site, j'ai respecté les standards du W3C et de Google. Tout d'abord, j'ai effectué des tests sur le site <https://validator.w3.org> afin d'examiner mon code html et de corriger les éventuels erreurs.

J'ai aussi réalisé un test via **Lighthouse** qui permet notamment de mesurer les performances du site, l'accessibilité, les bonnes pratiques et le SEO, c'est ce dernier qui va nous intéresser ici.

(Nous reviendrons sur le reste dans la partie « Performance »)



Le test SEO n'était pas à 100% dès le début, mais grâce aux recommandations est au système d'audit de Lighthouse j'ai pu augmenter mon score de SEO.

J'ai commencé par mettre en place une méta description dans le Header.

J'ai fait attention à l'utilisation des balises « **h** » pour les titres, un seul **h1** et un ordre logique pour les suivantes.

J'ai ajouté aussi des descriptions dans chaque image pour qu'elles soient indexées par le moteur de recherche par le biais d'une balise « **alt** ».

The screenshot shows the Lighthouse SEO audit results for a website. The overall score is 100, displayed prominently at the top. Below the score, the category "SEO" is listed. A note states: "These checks ensure that your page is optimized for search engine results ranking. There are additional factors Lighthouse does not check that may affect your search ranking. [Learn more](#)". Under the "Passed audits" section, there are 13 items listed, each with a green circular icon indicating success. The items include: Has a `<meta name="viewport">` tag with `width` or `initial-scale`, Document has a `<title>` element, Document has a meta description, Page has successful HTTP status code, Links have descriptive text, Links are crawlable, Page isn't blocked from indexing, Image elements have `(alt)` attributes, Document has a valid `hreflang`, Document has a valid `rel=canonical`, Document uses legible font sizes — 98.3% legible text, Document avoids plugins, and Tap targets are sized appropriately — 100% appropriately sized tap targets.

J'ai mis en place la **meta viewport** pour l'accessibilité au mobile.

J'ai installé une balise `<title>` dans chaque page du site.

Tout les liens ont des textes descriptifs et sont « crawlable » .

J'ai précisé la langue du site via la balise `<html lang="fr">`.

J'ai ajouté un URL canonique, plutôt important pour le référencement.

Et j'ai ajouté une favicon pour la plupart des navigateurs et une favicon pour apple.

Je passe un peu rapidement sur tout cela car j'aimerais exposé un peu plus en détail autre chose que j'ai mis en place.

J'ai notamment ajouter des balises meta Open Graph.

Open Graph est un protocole créé par Facebook qui permet d'optimiser le contenu d'une page web afin que les utilisateurs puissent la partager sur les principaux réseaux sociaux Facebook, Twitter, LinkedIn... Ce protocole facilite le partage depuis un ordinateur, le navigateur d'un smartphone ou une application mobile.

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="UTF-8">
    <title>{% block title %}Pont-Château Handball • Club de Handball • Site officiel du PCHB{% endblock %}</title>
    >
    <meta name="description" content="Pont-Château Handball est un club de Handball de Loire-Atlantique regroupant des équipes du Baby-Hand aux Séniors Masculines.">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="canonical" href="https://www.pchb.quentin-andreani.fr/public/" />
    <link rel="shortcut icon" type="image/png" href="{{ asset('logoPCHB64.png') }}"/>
    <link rel="apple-touch-icon" href="{{ asset('logoPCHB128.png') }}"/>

    <meta property="og:type" content="article" />
    <meta property="og:url" content="http://www.pontchateau-handball.fr/" />
    <meta property="og:site_name" content="Pont-Château Handball" />
    <meta property="og:title" content="Pont-Château Handball" />
    <meta property="og:description" content="Pont-Château Handball est un club de Handball de Loire-Atlantique regroupant des équipes du Baby Hand aux Séniors Féminines et Masculines. Ce club de sport à Pont-Château permet la pratique du Handball des plus jeunes jusqu'au plus grand ! Rejoignez le club de Pont-Château Handball. Vous pouvez aussi suivre la page de Pont-Château Handball ..." />
    <meta property="og:image" content="logoPCHB128.png" />
```

Voici la plupart des balises cité plus haut et mise en place dans le `<head>` qui favorisent le référencement sur les moteurs de recherches et les réseaux sociaux.

```
<div class="navLogo">
  <a href="{{ path('home') }}></a>
</div>
```

Ici nous avons l'exemple d'une description d'une image dans le paramètre « alt ».

Il est aussi important de préciser que j'ai mis en place un sitemap.xml qui est généré automatiquement via une mise en place d'un SitemapController dans Symfony, ainsi lorsqu'une nouvelle page est créée (nouvelle équipe, ou nouvel article par exemple) l'url est générée automatiquement dans le sitemap.xml :

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9" xmlns:xhtml="http://www.w3.org/1999/xhtml" xmlns:image="http://www.google.com/schemas/sitemap-image/1.1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd">
  <url>
    <loc> https://www.pchb.quentin-andreani.fr/public/ </loc>
  </url>
  <url>
    <loc> https://www.pchb.quentin-andreani.fr/public/resultat </loc>
  </url>
  <url>
    <loc> https://www.pchb.quentin-andreani.fr/public/login </loc>
  </url>
  <url>
    <loc> https://www.pchb.quentin-andreani.fr/public/plannings </loc>
  </url>
  <url>
    <loc> https://www.pchb.quentin-andreani.fr/public/articles/actualites </loc>
  </url>
  <url>
    <loc> https://www.pchb.quentin-andreani.fr/public/organigramme/organigramme </loc>
  </url>
  <url>
    <loc> https://www.pchb.quentin-andreani.fr/public/contacts/new </loc>
  </url>
  <url>
    <loc> https://www.pchb.quentin-andreani.fr/public/inscription/1 </loc>
  </url>
  <url>
    <loc> https://www.pchb.quentin-andreani.fr/public/historique/1 </loc>
  </url>
  <url>
    <loc> https://www.pchb.quentin-andreani.fr/public/articles/1 </loc>
    <lastmod>2021-02-02</lastmod>
    <image:image>
      <image:loc>https://www.pchb.quentin-andreani.fr/uploads/images/MjAxOTEyMzY2Tc2YWIS5MjIiMDM3MzIxNWUwZjNkZTU4ZjY5OGY-6019b3d8a2bb1.jpg</image:loc>
      <image:title>Senior Masculin</image:title>
    </image:image>
  </url>
  <url>
    <loc> https://www.pchb.quentin-andreani.fr/public/articles/2 </loc>
    <lastmod>2021-02-03</lastmod>
    <image:image>
      <image:loc>https://www.pchb.quentin-andreani.fr/uploads/images/MjAxOTRwZWQ4NDY2NzR1Y1c0NDAwY1N1OTgwZTR2MTn5YTYwXGM-601a9fd11976.jpg</image:loc>
```

Je me suis beaucoup intéressé à la manière de pouvoir générer des « **Rich Snippets** » que le club possède déjà lorsque on recherche directement « Pontchâteau handball » sur Google :

www.pontchateau-handball.fr ▾

Pont-Château Handball • Club de Handball • Site officiel du ...

Pont-Château Handball est un club de Handball de Loire-Atlantique regroupant des équipes du Baby Hand aux Séniors Féminines et Masculines.

Vous avez consulté cette page de nombreuses fois. Date de la dernière visite : 15/03/21

Planning des entraînements

Pont-Château Handball est un club de Handball de Loire ...

[Autres résultats sur pontchateau-handball.fr »](#)

Plannings des Matchs

Plannings des Matchs. Retrouvez ici les plannings des matchs ...

J'ai découvert que ces « Rich Snippets » pouvaient être générés grâce aux **données structurées** de Google. Ces données structurées sont un format normalisé permettant de fournir des informations sur une page et de classer le contenu de cette page. Elles peuvent être générées via un script Json-LD.

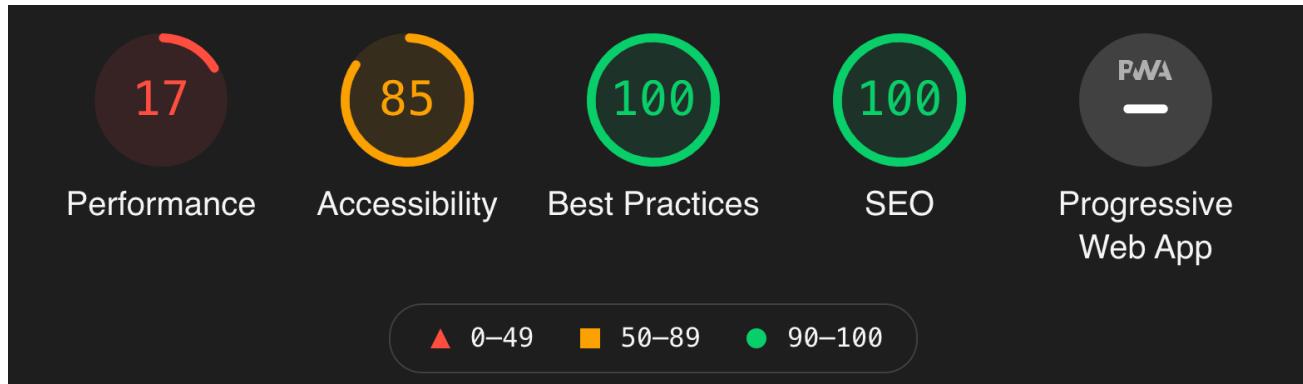
J'ai donc moi-même créé un script Json en suivant les recommandations de Google sur le « **Google Search Central** » afin de générer un fil d'ariane avec le **@type:BreadcrumbList** :

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "BreadcrumbList",
  "itemListElement": [
    {
      "@type": "ListItem",
      "position": 1,
      "name": "Actualité du club",
      "item": "https://www.pchb.quentin-andreani.fr/public/articles/actualites"
    },
    {
      "@type": "ListItem",
      "position": 2,
      "name": "Résultats",
      "item": "https://www.pchb.quentin-andreani.fr/public/resultat"
    },
    {
      "@type": "ListItem",
      "position": 3,
      "name": "Contactez-Nous",
      "item": "https://www.pchb.quentin-andreani.fr/public/contacts/new"
    },
    {
      "@type": "ListItem",
      "position": 4,
      "name": "Plannings",
      "item": "https://www.pchb.quentin-andreani.fr/public/plannings"
    },
    {
      "@type": "ListItem",
      "position": 5,
      "name": "Inscription",
      "item": "https://www.pchb.quentin-andreani.fr/public/inscription/1"
    },
    {
      "@type": "ListItem",
      "position": 6,
      "name": "Organigramme",
      "item": "https://www.pchb.quentin-andreani.fr/public/organigramme/organigramme"
    }
  ]
}</script>
```

Les données structurées dans un script Json-LD pour l'instant avec les liens de mon sous-domaine mais que je modifierais au moment de la mise en production sur le domaine du club.

TEST DE PERFORMANCE

Pour vérifier la rapidité d'affichage, l'accessibilité, les bonnes pratiques au niveau du code ainsi que le référencement, j'ai utilisé le module **Lighthouse** cité un peu plus haut, pour noter un évolution vis à vis de la performance, j'ai réalisé un premier test de performance en version mobile :



Les performances n'étant clairement pas convaincante, en tout cas pour la version mobile, j'ai décidé de suivre les recommandations de Google afin d'augmenter mon score de performance ainsi que de l'accessibilité.

Pour commencer j'ai éliminé de nombreuse image de test, qui, je me suis rendue compte n'était pas nécessaire. J'ai ensuite compresser ces images via un site très pratique qui s'appelle "[TinyJpg.com](#)". Une fois la compression établie j'ai ensuite convertie mes images jpg et png en image au format **WebP**. Le **WebP** est un format d'image matricielle développé et mis à disposition du public par Google, il exploite un algorithme de compression avec pertes prédictif.

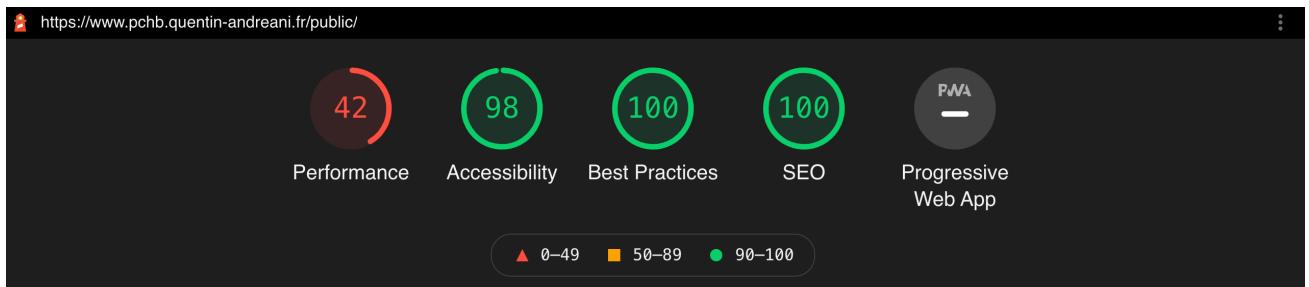
Il me fallait ensuite régler le problème des images « hors écran » ainsi que des iframes qui prenaient beaucoup de ressources lors du chargement de la page. J'ai découvert qu'il était possible de différer leurs chargements mais j'ai du faire quelques recherches pour ça que je décrirais dans la partie de recherche en Anglais.

J'ai découvert qu'il était possible d'ajouter un simple attribut dans les balise **** et **<iframe>** il s'agit du l'attribut « **loading** » avec la valeur « **Lazy** ». Cet attribut est depuis peu nativement pris en charge par les navigateurs Chromium ainsi que Firefox, Safari est en cours de déploiement.

```
<iframe title="Classement Séniors" id="600ff978503df30a15f1e9df" height="499" src="https://scoreenco.com/widget/600ff978503df30a15f1e9df/" style="display: block; width: 100%; overflow: auto; margin: auto; border-width: 0px;" loading="lazy"></iframe>
```

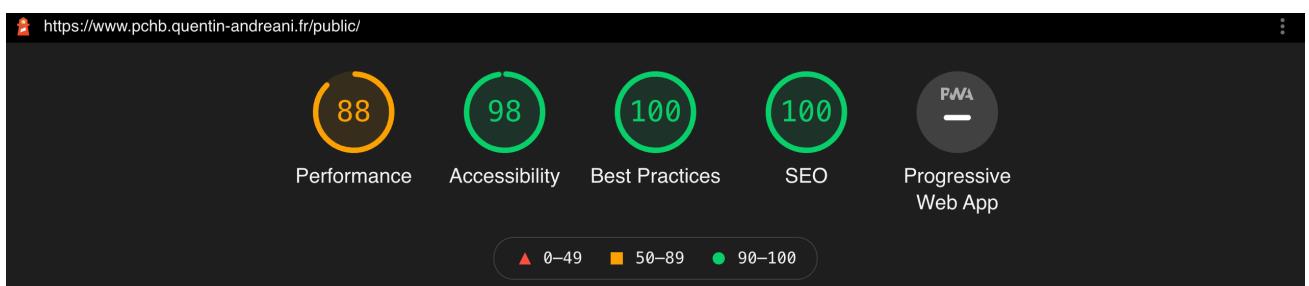
L'attribut loading avec la valeur lazy pour différer le chargement de l'iframe

Suite à cela, voici les valeurs que j'ai réussi à obtenir :



Les nouvelles valeurs obtenu pour la version mobile

Et voici les valeurs obtenu pour la version Desktop :



Les valeurs obtenus pour la version Desktop sont plutôt bonne mais je travail toujours sur la version mobile afin d'améliorer au mieux le score de performance.

Et de manière général je travail toujours sur la performance du site dans son ensemble afin d'avoir une expérience utilisateur optimal.

RECHERCHE EN ANGLAIS

Dans un désir de vouloir optimiser ma page web afin d'améliorer mon score de Performance sur Lighthouse, il me fallait réduire mon temps de chargement, c'est ainsi qu'après quelques recherches j'ai découvert l'attribut mentionné plus haut « loading ».

La plupart des ressources trouvé n'était pas très précise et il me fallait des informations de meilleures qualités ainsi que la manière de le mettre en place.

C'est ainsi que je suis tombé sur **la page « Lazy loading » de Mozilla**, cette page est très intéressante et contenait les informations que je désirais avec une traduction de ma part au préalable, car malgré que ce soit une documentation Mozilla, celle-ci n'est pas disponible en Français pour le moment.

La traduction m'a permis de mettre en place le chargement différé sur les images hors écran ainsi que des iframes :

JE SUIS TOMBÉ SUR CE PARAGRAPHE :

Images and iframes

Very often, webpages contain many images that contribute to data-usage and how fast a page can load. Most of those images are off-screen ([non-critical](#)), requiring user interaction (an example being scroll) in order to view them.

Loading attribute

The [loading](#) attribute on an [``](#) element (or the [loading](#) attribute on an [`<iframe>`](#)) can be used to instruct the browser to defer loading of images/iframes that are off-screen until the user scrolls near them.

```

<iframe src="video-player.html" title="..." loading="lazy"></iframe>
```

The [load](#) event fires when the eagerly-loaded content has all been loaded; at that time, it's entirely possible (or even likely) that there may be lazily-loaded images that are within the [visual viewport](#) that haven't yet loaded.

QUE J'AI TRADUIS PAR :

Images et Iframes

Très souvent, les pages Web contiennent de nombreuses images qui contribuent à l'utilisation des données et à la vitesse de chargement d'une page. La plupart de ces images sont hors écran (non-critique), nécessitant une interaction de l'utilisateur (un exemple étant le scroll ou défilement) afin de les voir ou de les visualisés.

Attribut de chargement

L'attribut `loading` sur l'élément `` (ou l'attribut `loading` sur un `<iframe>`) peut être utilisé pour demander au navigateur de différer le chargement des images / iframes qui sont hors écran jusqu'à ce que l'utilisateur les fasse défiler.

```

<iframe src="video-player.html" title="..." loading="lazy"></iframe>
```

L'événement de chargement se déclenche lorsque le contenu chargé avec « empreusement » à été chargé; à ce moment-là, il est tout à fait possible (ou même probable) qu'il y ait des images chargées « paresseusement » qui se trouvent dans l'écran visible qui ne sont pas encore chargées.

VIE PRIVÉE

Afin de mettre en place un système de gestion de cookie, j'ai décidé d'utiliser « tarteaucitron » qui est un script javascript Open-Source qui gère les cookies de façon automatique et conforme aux recommandations du RGPD.

Il suffit de mettre en place deux script dans notre page principale et le plugin se charge de trouver les cookies à activer ou non sur le site :

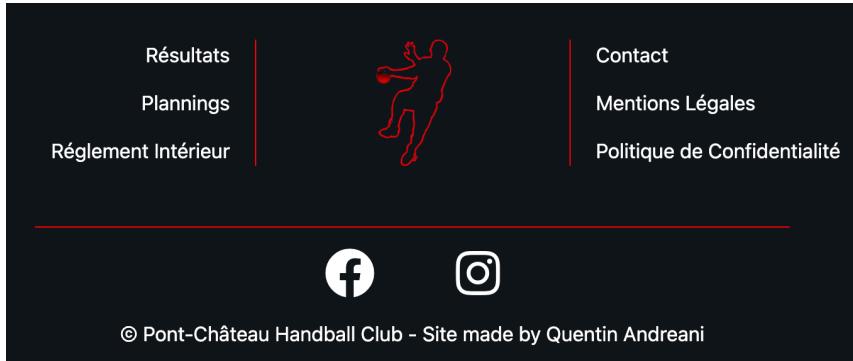
```
<script type="text/javascript" src="{{asset('tarteaucitron/tarteaucitron.js')}}"></script>
```

The screenshot shows a dark-themed cookie consent interface. At the top, a banner states: "Ce site utilise des cookies et vous donne le contrôle sur ceux que vous souhaitez activer". It includes three buttons: "✓ Tout accepter" (green), "✗ Tout refuser" (red), and "Personnaliser". Below this, the title "Panneau de gestion des cookies" is centered. A message reads: "En autorisant ces services tiers, vous acceptez le dépôt et la lecture de cookies et l'utilisation de technologies de suivi nécessaires à leur bon fonctionnement." A "Préférences pour tous les services" section has buttons for "✓ Tout accepter" and "✗ Tout refuser". The "Cookies obligatoires" section states: "Ce site utilise des cookies nécessaires à son bon fonctionnement qui ne peuvent pas être désactivés." It has a "✓ Autoriser" button. The "+ Mesure d'audience" section lists "Google Analytics (gtag.js)" and notes it can deposit 10 cookies. It has buttons for "✓ Autoriser" and "✗ Interdire". A link "En savoir plus - Voir le site officiel" is provided.

Voici comment se présente la gestion des cookies sur le site, il permet à l'utilisateur de sélectionner ou non ce qu'il souhaite stocker.

POLITIQUE DE CONFIDENTIALITÉ

J'ai aussi mis en place une politique de confidentialité accessible dans le footer sur toutes les pages du site :



Le footer du site avec le liens vers les mentions légales et la politique de confidentialité.

Dans la page de confidentialité il est précisé tout les droits informatiques et liberté pour les utilisateurs, voici un extrait de la politique de confidentialité :

« 5. Droits Informatique et Liberté

Vous disposez des droits suivants concernant vos Informations Personnelles, que vous pouvez exercer en nous écrivant à l'adresse postale mentionnée au point 1 ou en remplissant le formulaire ci-contre.

Droit d'accès et de communication des données Vous avez la faculté d'accéder aux Informations Personnelles qui vous concernent.

Cependant, en raison de l'obligation de sécurité et de confidentialité dans le traitement des données à caractère personnel qui incombe au HandBall Club de Pontchâteau, vous êtes informé que votre demande sera traitée sous réserve que vous rapportiez la preuve de votre identité, notamment par la production d'un scan de votre titre d'identité valide (en cas de demande par notre formulaire électronique dédié) ou d'une photocopie signée de votre titre d'identité valide (en cas de demande adressée par écrit).

le HandBall Club de Pontchâteau vous informe qu'il sera en droit, le cas échéant, de s'opposer aux demandes manifestement abusives (de par leur nombre, leur caractère répétitif ou systématique).

Pour vous aider dans votre démarche, notamment si vous désirez exercer votre droit d'accès par le biais d'une demande écrite à l'adresse postale mentionnée au point 1, vous trouverez en cliquant sur le lien suivant un modèle de courrier élaboré par la Commission Nationale de l'Informatique et des Libertés (la « CNIL »). <https://www.cnil.fr/fr/modele/courrier/exercer-son-droit-dacces>

CONCLUSION

Ce rapport m'a permis d'exposer tout les étapes de créations de mon projet pour l'association Pontchâteau Hand-Ball. Ce projet m'a permis de mettre en avant les activités sportives du club avec un environnement utilisateur dédié.

Les trois grandes étapes de mon projet consistaient en premier lieu d'établir les besoins avec l'association pour que nous puissions réaliser le cahier des charges.

Dans un second temps, connaître et déterminer les besoins fonctionnels du site. Pour ce faire j'ai réalisé des diagrammes me permettant d'élaborer l'interface utilisateur et d'établir les différents rôles.

Pour finir, la dernière étape était la conception du site, en prenant en compte les aspects de sécurité, d'ergonomie, de responsive et de test.

Mon projet a été une expérience très enrichissante malgré, les difficultés et les contraintes de temps, je garde une impression très positive de ce projet.

Le fait d'avoir suivie toutes les étapes de réalisation d'un site, ainsi que d'avoir réalisé ce site seul et dans mes propres contraintes, dans ma propre organisation, m'a permis de voir beaucoup plus clair vis à vis d'une conception similaire dans le monde professionnel, le fait de me diriger seul sur le projet en rencontrant chaque difficulté et m'en-sortir toujours plus grandi.

Grâce à cette première expérience, j'ai maintenant une vision plus précise de ce que peuvent donner des projets en entreprise.

Le club est satisfait du rendu final du nouveau site, malgré cela il me reste encore beaucoup à faire, au niveau des fonctionnalités, je n'ai pas pu mettre en place tout ce que j'aurais voulu, comme par exemple une solution de e-commerce, d'inscription au club en ligne ainsi que de paiement en ligne mais il ne s'agit que d'une question de temps.

Je me suis engagé auprès d'eux afin de leur apporter ces améliorations et fonctionnalités, mise de côté par manque de temps.

Il me tarde de me remettre au travail et d'apporter cette solution à l'association !

Pour la suite, je ne compte absolument pas m'arrêter là, je suis déterminé à améliorer mes compétences en Php et Javascript et passer le diplôme de concepteur développeur d'application !

Je souhaite aussi élargir mes connaissances en cybersécurité, car je sais qu'actuellement je n'ai fait qu'effleurer la surface visible de l'iceberg d'un domaine qui me paraît indispensable dans le monde d'aujourd'hui.

Cette formation a été très bénéfique pour moi, et m'a fait devenir un passionné qui n'est pas prêt de s'arrêter de coder.