



2020

REALISATION DU SITE WEB « LES LAMES DU PONANT »

Gwendal Bescont

TABLE DES MATIERES

Avant-Propos	3
Remerciement.....	3
Liste des compétences couvertes par le référentiel	3
Résumé du projet.....	4
Environnement de travail	4
Méthodologie.....	Erreur ! Signet non défini.
Les outils utilisés	5
Back-end :	5
Front-end :	5
Logiciels :	5
Veille	6
Cahier des charges	7
Expression des besoins	7
Specifications fonctionnelles.....	8
Arborescence détaillée	9
Code couleur	10
Logo.....	11
Police d'écriture	11
Maquetter une Application	12
Realisation du site	13
Planification du projet.....	13
Définition des Rôles	14
Digrammes	15
MCD	15
UML.....	16
Diagramme de séquence enregistrement de compte.....	17
Diagramme de séquence connexion au site.....	18

Digramme de séquence envoi de message sur le forum.....	18
Developpement Back end	19
Création du squelette et de la base du projet	19
Construction de la base de donnée	19
Création Des Enties.....	20
Utilisateur et sécurité.....	21
Veille sur la sécutrité.....	23
Exemple de code.....	27
Le forum.....	27
Requêtes Ajax.....	30
Réaliser une interface utilisateur avec gestion de contenu	31
Suppression d'une rubrique	32
Ajout/Maj d'une rubrique	32
Front End.....	33
Mise en place du Framework CSS	33
Sass et JS	34
Réaliser une interface web statique et dynamique	35
Affichage mobile	35
Mise en production.....	37
Recherche en anglais	38
Le référencement.....	39
Test de performance.....	39
Vie privée	40
Conclusion.....	41

AVANT-PROPOS

Ce dossier Professionnel a été réalisé dans le cadre d'une reconversion professionnelle à la suite d'un accident du travail. J'ai eu l'opportunité de bénéficier d'un financement dans le cadre d'un « CIF » auprès de Transition Pro Nantes qui m'a aiguillé dans mon choix afin d'exercer dans le secteur du Web.

J'ai entamé ma formation début avril 2020 en télétravail en pleine période 'Covid' qui a impacté le fait d'être en présentiel au sein du centre de formation mais n'a aucunement affecté la qualité de l'enseignement, ma motivation ainsi que ma détermination .

REMERCIEMENT

Durant ma formation chez Arinfo Saint-Nazaire nous avons été formés par Nicolas Gicquel, dont nous étions la première promo je tenais fortement à le remercier pour le temps passé à nos côtés ainsi que sa patience qui a amplement le mérite d'être saluée. Je tenais aussi à remercier tout le personnel de Arinfo Saint-Nazaire dont l'accueil a été plus que chaleureux et m'a donné envie de venir étudier tous les jours.

Je tenais aussi à remercier tous les créateurs de tutoriels sur internet qui m'ont permis de booster mes apprentissages et aussi de m'apporter des clefs pour élaborer mon projet, je pense à « Grafikart, Open Classroom, Lior Chamla ainsi que nouvelles technos ».

LISTE DES COMPETENCES COUVERTES PAR LE REFERENTIEL

- 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 ou web mobile
- 9 - Elaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce

RESUME DU PROJET

L'association Les Lames Du Ponant pratiquant les AMHE (arts martiaux historique européens) existant depuis 2013, souhaite avoir un site vitrine dans le but d'apporter un nouveau support de communication. Des discussions ont été entamées début juillet afin d'évaluer la charte graphique ainsi que les différents éléments qui devront y être présents.

Après quelques réunions, nous avons pu établir sur les grandes lignes l'aspect fonctionnel du site qui devra permettre et partant de ses éléments j'ai pu établir un cahier des charges plus élaboré côté développement.

Une veille concurrentielle a été faite afin d'apporter des idées et des solutions pendant toute la rédaction afin que le projet corresponde au mieux aux attentes de l'association.

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

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

Etablissement des différents rôles utilisateur avec pour chaque un détail complet des différents accès.

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

Côté back-end le site sera codé avec le framework mvc Symfony et en front-end avec le framework CSS Materialize.

Durant le processus de développement il fut décidé d'opter pour une création de site web en méthode dites « AGILE » afin de pouvoir échanger sur le projet et lui apporter des modifications dans le cas échéant.

Pour la partie communication, nous avons validé l'utilisation de la plateforme de discussion libre « **Discord** », qui permet de correspondre en tchat écrit comme oral et aussi de faire du partage d'écran.

Une réunion hebdomadaire a été mise en place afin d'élaborer la création du site et lui assigner les différents éléments par page. J'ai endossé différentes casquettes afin de correspondre au mieux aux attentes de l'association que je considérais comme un client et j'ai essayé d'être le plus flexible possible, ce qui m'a confronté à plusieurs problèmes d'intégration et de découpage du site. Je prendrais le temps de m'étendre sur le sujet dans « **la conclusion** » à la fin de mon dossier.

Voici les différents outils, logiciels, Framework utilisés pendant le développement. J'ai utilisé pour l'essentiel des licences **Open Source**.

BACK-END :

- « **Symfony** » qui est un Framework Open Source MVC (*modèle-vue-contrôleur*) basé sous le langage PHP (*Hypertext Preprocessor*).
- Concernant la gestion de base de données j'ai utilisé « **MYSQL** » (*My Structured Query Language*) qui est un gestionnaire Open-Source couplé d'une interface simple et facile d'accès.
- Durant le processus de développement j'ai été amené à utiliser l'outil « **WAMP serveur** » qui a des fins d'essais, m'a permis d'afficher et tester mes scripts PHP sous une interface locale et aussi afficher le site en live afin de travailler sur la mise en page.

FRONT-END :

- « **Materialize** » qui est un Framework CSS (*cascading Style Sheet*) à la fois épuré et moderne. Pour la partie CSS j'ai utilisé le préprocesseur CSS « **SASS** » (*Syntactically Awesome Stylesheets*). Il permet de mieux gérer son code en CSS et facilite les mises à jour en cas de modification du code par exemple avec des **Mixins**.

LOGICIELS :

- Concernant mon environnement de développement je me suis tournée vers l'ide **VS Code** pour lequel un très large panel d'extensions existe et facilite grandement la lecture du code avec une aide non négligeable pour certaines d'entre elles lorsqu'il s'agira d'écrire des requêtes ou créer des fonctions dans les contrôleurs.
- Coté terminal j'ai utilisé **powershell** dans sa dernière version, couplé à la gestion de dépendance **composer** il m'a permis la construction de mon projet, le téléchargement des différents bundles ainsi que d'effectuer les mises à jour nécessaire
- Le Projet est synchronisé sur la plateforme « **GitHub** » afin de faciliter la programmation avec la permission de revenir en arrière au niveau du code en cas de problème par le biais de « **Branch** » qui correspondent à un clone du projet « **Branch Master** », c'est-à-dire le dossier projet validé et sans bugs sur lequel on peut se greffer tout en travaillant dans un environnement sécurisé.
Cela m'a permis durant ma phase de développement d'en créer plusieurs avec pour chacune un thème bien particulier afin de découper au mieux mon projet et ainsi effectuer des tests sans prises de risques.
GitHub offre aussi la possibilité de partager son code sur plusieurs ordinateurs avec un système de '**clone**' ce qui m'a énormément servi lors de mes premières phases de développement et ma facilité l'échange en cas de problème auprès de mon tuteur qui pouvait y accéder à n'importe quel moment.

Dans le but de correspondre au mieux aux attentes de l'association, j'ai entamé différentes recherches afin de trouver des renseignements sur divers sites traitant du thème médiéval, des AMHE.

Le fruit de mes recherches m'a grandement aidé lors de mes réunions et a facilité mes échanges auprès de l'association et les rassurait sur ma motivation à entreprendre ce projet.

Ces recherches m'ont avant tout permis d'apprendre ce que sont les AMHE et aussi trouver des informations pertinentes que je pouvais intégrer au projet. Voici les différents sites qui m'ont énormément renseigné sur l'escrime ainsi que l'univers médiéval en général :

- Fédération Française des AMHE : <https://www.ffamhe.fr>
- Site web des AMHE de Nantes : <http://www.escrimeurs-libres.fr>
- Site web De l'association des AMHE de Montpellier : <http://chevaleresque.free.fr>
- Site web détaillant les AMHE dans leur aspect historique : <https://www.amheonweb.net>

Site web traitant sur la reconstruction médiévale :

- <http://www.federation-francaise-medievale-et-renaissance.fr>
- <https://www.degueulesetdargent.fr>



Voici un exemple du site intitulé de gueules et d'argent, partagé lors d'une réunion par l'un des membres de l'association qui a fortement inspiré le design final du projet.

EXPRESSION DES BESOINS

L'association voulant se développer, il lui fallait un site vitrine lui permettant de faire connaître ses activités afin de manière plus élaborée que via Facebook et Instagram. Il devra implémenter plusieurs rôles utilisateur (détaillé dans la partie rôle) et devra être modulable sur la forme ainsi que sur le fond avec une gestion de contenu.

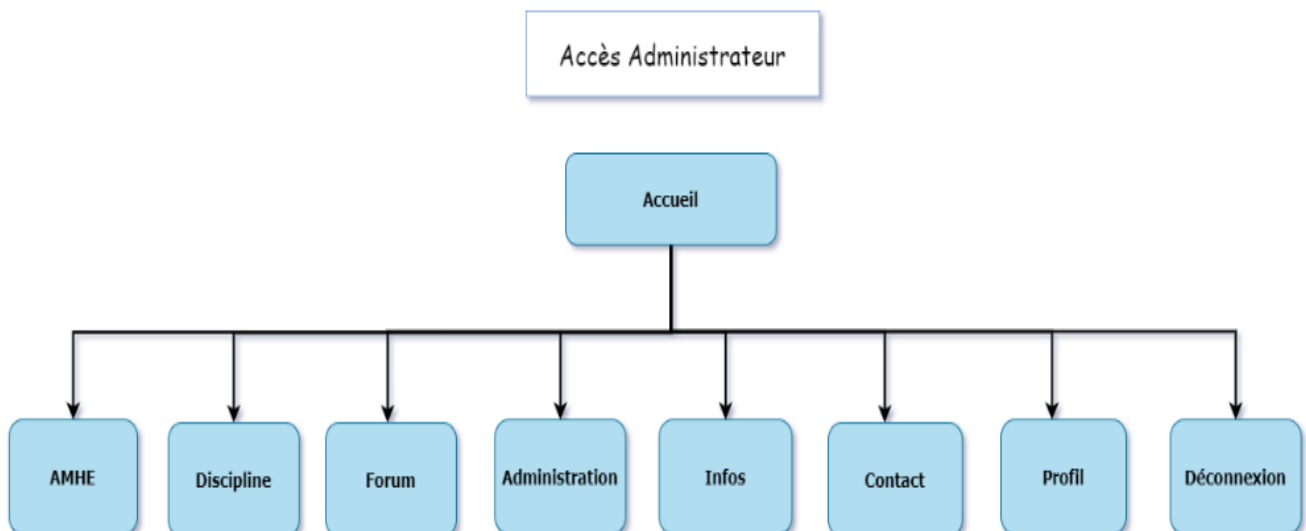
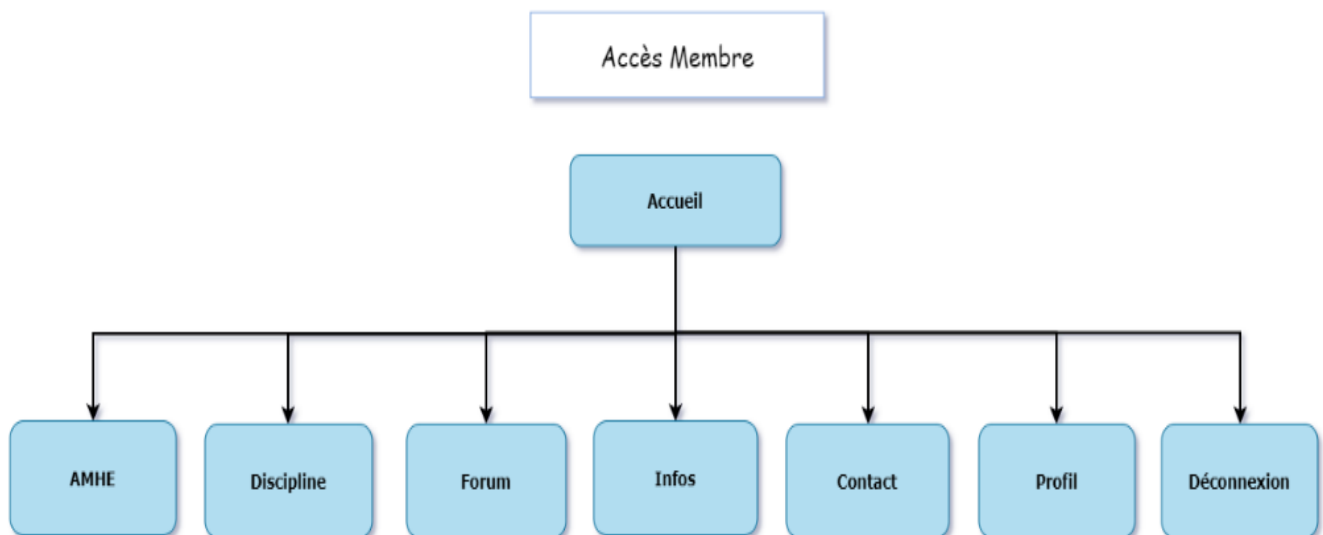
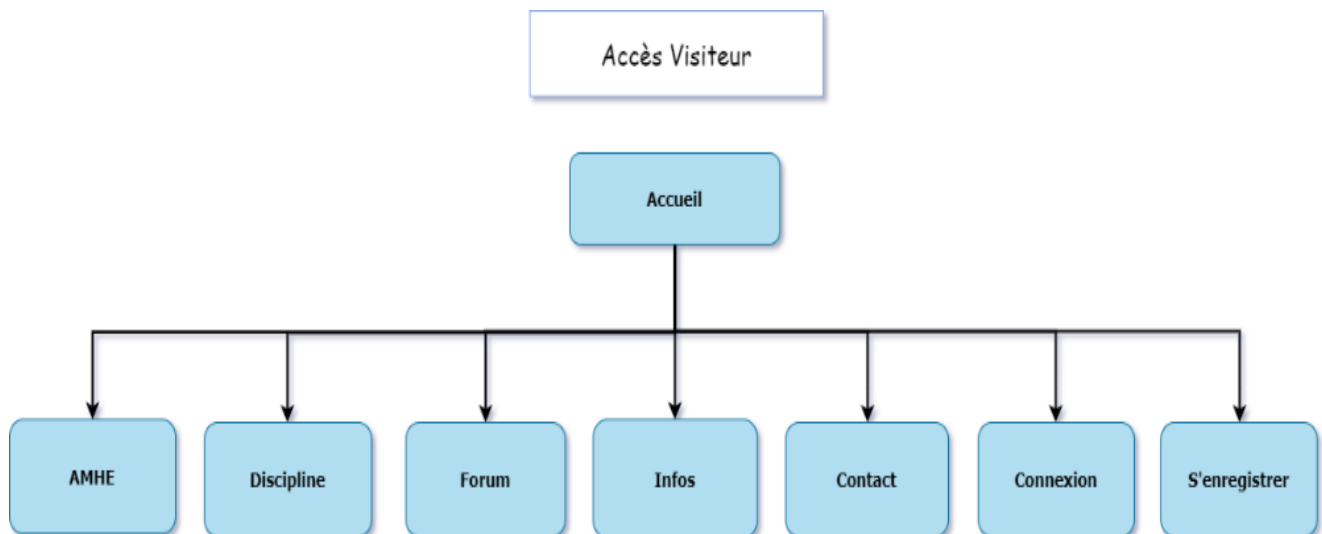
Voici les demandes principales concernant le site :

- L'inscription de nouveaux membres
- Permettre de faire connaître l'association ainsi que les AMHE en mettant en avant des techniques de SEO
- Apporter une visibilité sur les futurs évènements à venir
- Mettre en avant les différents modules d'entraînement : « épée longue, épée bocle, combat viking, etc...
- Incorporer un système de login mot de passe pour l'accès utilisateur et administrateur
- Mettre en place d'un forum afin d'apporter un lieu d'échange
- Mettre en place d'un formulaire de contact
- Mettre un système d'administration, permettant la gestion des contenus

Pour la partie visuelle du site :

- Le site devra avoir la couleur bleue comme dominante
- Il devra contenir de la parallaxe sur la page principale
- Il devra comporter des slider pour la partie discipline

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



Chaque page du site est composée d'une à plusieurs rubriques dont voici les principaux éléments :

- **Page Accueil**

- Présentation succincte de l'association
- Carousel représentant les disciplines avec un lien cliquable renvoyant vers la page discipline
- Les actualités permettant d'informer les visiteurs et membres des différentes alertes/ événements
- Information pratique, affichant les informations essentielles afin d'avoir les horaires ainsi que le lieu d'entraînement

La page d'accueil est très importante car elle répond à elle seule à la question « [qui, quoi et où ?](#) » et permet à un visiteur d'avoir accès aux informations essentielles sans avoir à naviguer sur toutes les pages du site. Sachant que l'utilisation mobile a une part importante au niveau du web (+ **222%** en 7 ans soit **53.3%** du trafic actuel) il fallait que les informations soient lisibles et accessibles rapidement en quelques « scroll ».

- **Page Discipline**

- Carousel représentant trois image en rapport avec la discipline concernée
- Une présentation de la discipline avec une image de l'instructeur

- **Page AMHE**

- Présentation de la FFAMHE (Fédération Française des Arts Martiaux Historiques Européens)
- Présentation des différentes Familles Médiévales
- Présentation détaillée de l'association Des Lames Du Ponant

- **Page Forum**

- Une page d'accueil
- Une page affichant les sujets
- Une page permettant de commenter un sujet

- **Page Profil**

- Edition de son profil en lui indiquant ses informations (pseudo, anniversaire...)
- Edition de son avatar

- **Page Infos**

- Affichage des horaires
- Implémentation d'une carte Open Street Map
- Adhésion et coût de l'activité
- FAQ (foire aux questions)

- **Page Connexion**

- Formulaire de connexion
Lien de récupération de mot de passe

- **Page Inscription**

- Formulaire avec demande d'email et vérification par double mot de passe

- **Page Administration**

- Possibilité de modifier toutes les pages du site
- Gestion des Section, sujets et messages du forum
- Gestion des utilisateurs
- Accès à la messagerie de la page contact

- **Page Contact**

- Page permettant de contacter l'association pour différents renseignements et aussi en cas de bugs sur le site

CODE COULEUR

Afin de correspondre au mieux aux t-shirts « bleu royal » de l'association, le site reprendra ce code couleur sur toutes les pages (il est à noter que la couleur bleue est la plus polyvalente et aussi la plus appréciée dans le monde), après plusieurs essais, voici les principales couleurs qui ont été retenues :



L'association avait plusieurs Logos dessinés par l'un des membres. Après quelques tests d'intégration nous avons décidé de garder celui qui s'accordait parfaitement avec la charte graphique du site.



POLICE D'ECRITURE

Deux polices d'écriture ont été définies pour tout le site.

Les titres seront écrits avec la police **Maven Pro**, elle correspond à la police d'écriture utilisée pour le **Logo** et permet de garder une homogénéité sur l'ensemble du site.

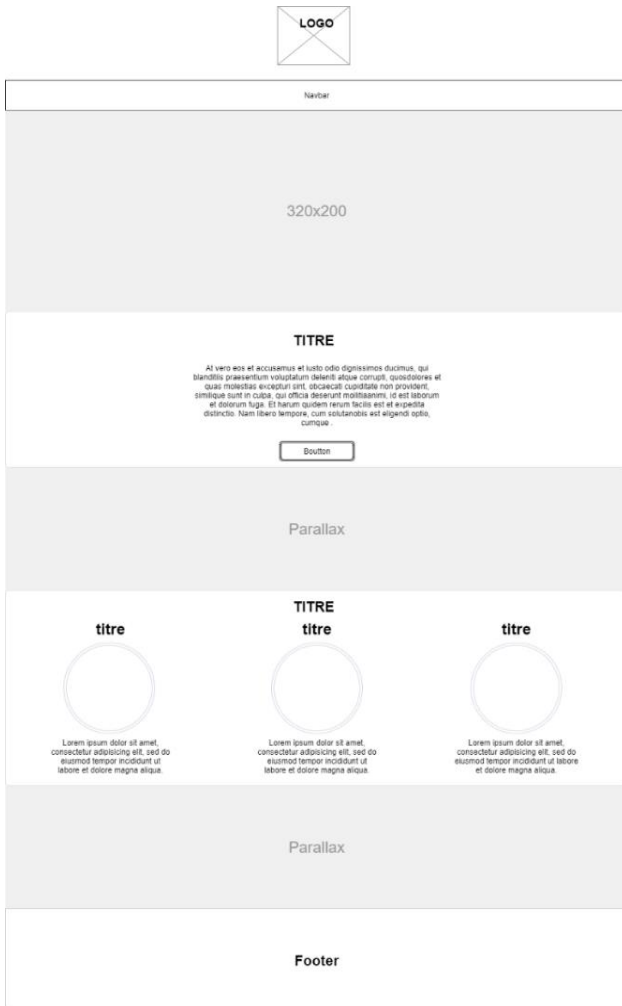
Penultimate
The spirit is willing but the flesh is weak
SCHADENFREUDE
3964 Elm Street and 1370 Rt. 21
The left hand does not know what the right hand is doing.

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.

Les autres textes type paragraphe, Span seront quant à eux écrit avec la police de caractère **ROBOTO** qui est une police de caractère commune utilisée pour l'interface Android et créée par **Christian Robertson**.

MAQUETTER UNE APPLICATION

Afin de réaliser la maquette du site des Lames Du Ponant, j'ai utilisé l'application gratuite **Draw.IO** qui permet de créer des wireframes ainsi que des diagrammes avec beaucoup de facilité.



Présentation du wireframe version « mobile »



Présentation du wireframe version « Desktop »



PLANIFICATION DU PROJET

J'aurais souhaité réaliser un diagramme de **Gant** afin de découper au mieux mon projet, étant novice et découvrant le Framework MVC Symfony j'étais dans l'incapacité de quantifier correctement le temps passé sur chaque rubrique. J'ai préféré établir une liste de tâches détaillée par ordre chronologique ci-dessous :

Back -end

1. Définir les rôles utilisateur.
2. Création de diagrammes.
3. Création du squelette Symfony.
4. Création de la base de données.
5. Création de entités avec leurs relations.
6. Mise en place du service global d'authentification utilisateur.
7. Sécurité du site et utilisateur.
8. Création de la partie administration du site.
9. Création du forum.
10. Mise en place gestion cookies, reCAPTCHA
11. Test de sécurité

Font end

1. Mise en place du framework CSS Materialize
2. Travail sur la partie header
3. Travail sur la partie Footer
4. Intégration des différents blocs en fonction des pages

Finalisation

1. Achat du nom de domaine
2. Passage en production
3. Phase de test auprès de l'association
4. Correction des bugs
5. Test du site en version mobile
6. Livraison du site

Avant de me focaliser pleinement sur la partie « **codage** » j'ai finalisé l'attribution des différents rôles utilisateurs pouvant parcourir le site. Il y aura **5 catégories** d'utilisateurs, les rôles modérateur, administrateur et super-administrateur seront directement créés en BDD tandis que chaque nouvel inscrit aura automatiquement le rôle user. Voici une présentation globale ainsi que les autorisations affiliées à chaque utilisateur :

LE VISITEUR

Le visiteur pourra naviguer sur l'ensemble du site sans aucun droit spécifique si ce n'est d'envoyer un message au modérateur ou super-administrateur via le formulaire de contact. Il peut devenir membre via l'onglet inscription du site.

LE MEMBRE/ ADHERENT

Le membre/adhèrent correspond à un visiteur qui s'est inscrit sur le site, il pourra accéder au forum, créer des sujets, poster des commentaires. Il a aussi la possibilité de devenir 'modérateur du forum' afin de faire respecter les règles de la FAQ.

LE MODERATEUR

Le modérateur pourra intervenir sur le site dans sa globalité mais restreint à certaines limites réservées à l'administrateur. Il sera en charge :

- De répondre aux sujets reçus dans l'onglet contactez nous
- D'ajouter du contenu à l'interface principale dans les blocsSection et catégorie du forum.
- Il aura la charge de modérer le forum

« Le site étant encore en développement, je n'ai pas incorporé ce rôle, cela sera fait lors d'une future mise à jour »

L'ADMINISTRATEUR

L'administrateur aura les plus hautes responsabilités du site. Il aura la charge :

- De maintenir le site, avec la capacité de modifier le code pour lui apporter de la stabilité
- De faire une refonte de certains modules du site
- D'administrer les utilisateurs dans leur globalité
- De pouvoir attribuer un rôle modérateur à un adhérent

LE SUPER-ADMINISTRATEUR (DEVELOPPEUR DU SITE)

Le super-administrateur aura les mêmes droits que l'administrateur et en plus aura accès à la messagerie contact sur les différentes alertes concernant les bugs du site afin de pouvoir les corriger. Il aura accès au serveur (détaillé dans la partie production), à la base de données ainsi qu'au code source du site.

*« Je détaillerais plus précisément la sécurité attribuée à chaque rôle utilisateurs par la suite dans la section **sécurité et utilisateurs** »*

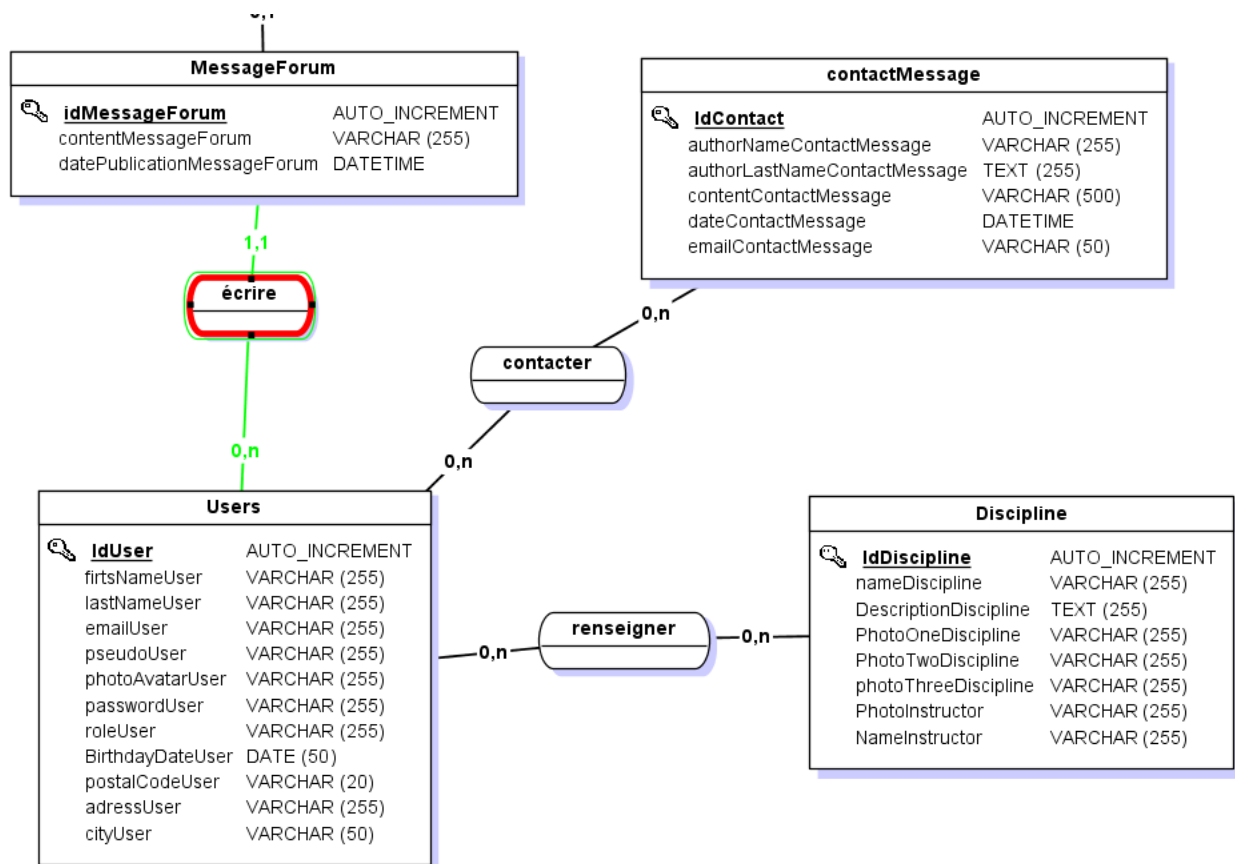
J'ai réalisé différents diagrammes ainsi qu'un MCD afin de me donner une feuille de route quant à la création des entités du site, leur contenu ainsi que les diverses relations qui pourront être mises en place.

Souhaitant démarrer le développement de mon projet en mettant en avant la partie utilisateur et sécurité, il me fallait un visuel détaillé de ces éléments que je devais implémenter au niveau de mon code ainsi que des bundles que j'allais utiliser pour arriver au résultat escompté.

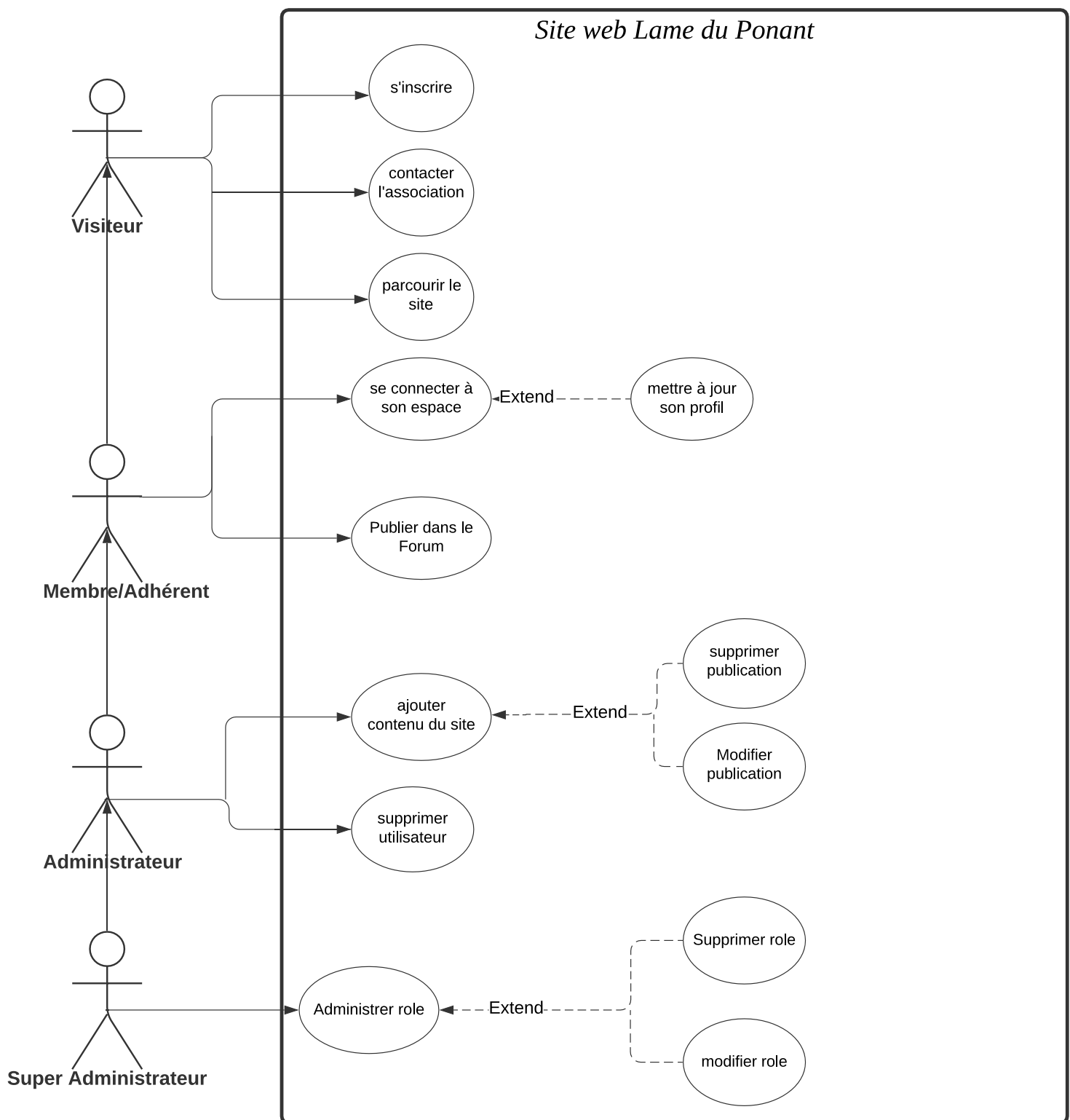
MCD

Pour construire mon MCD (Modèle Conceptuel des Données) j'ai utilisé le logiciel **J-MERISE** afin d'avoir une représentation schématique de mon projet. J'avais une visualisation globale de ma base de données et cela m'a permis de mettre en avant mes relations entre mes entités.

Voici un exemple du mcd que j'ai utilisé, on peut y voir les relations entre différentes tables :



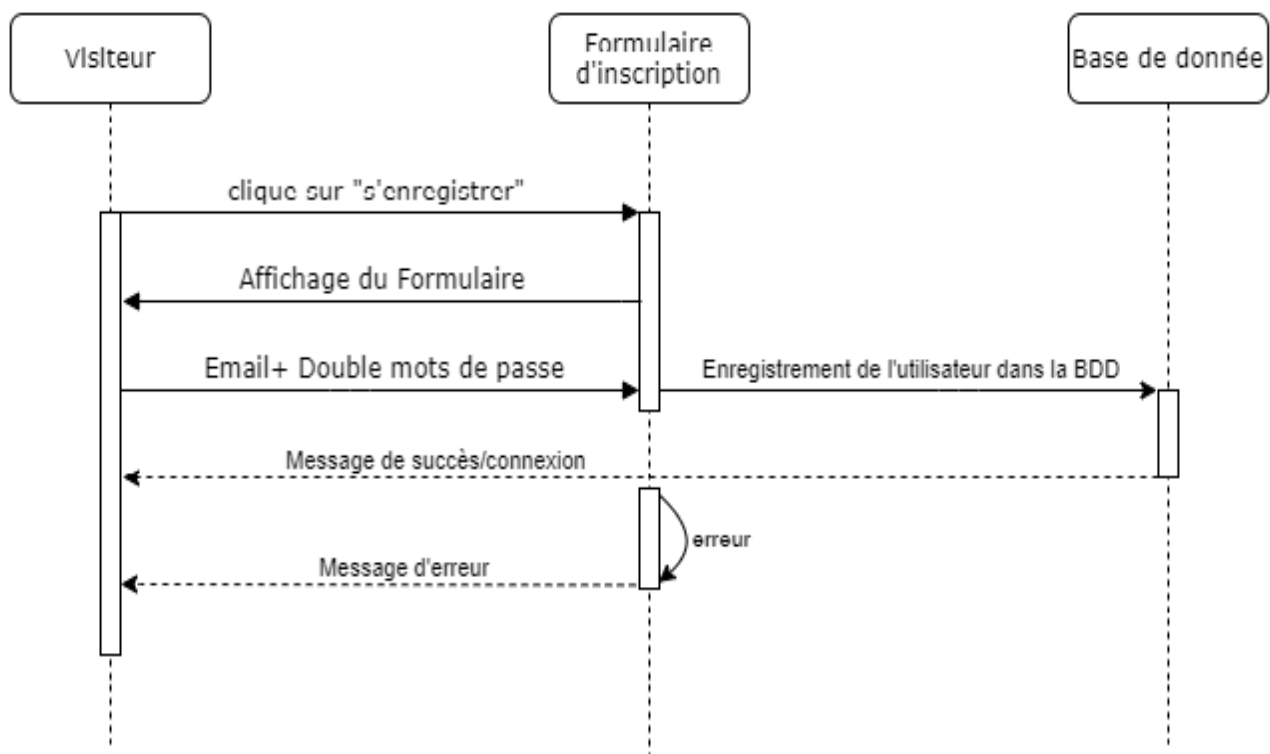
J'ai réalisé un diagramme de cas d'utilisation afin d'avoir une vision globale du comportement fonctionnel du site en fonction de chaque rôles utilisateur, en détaillant les niveaux d'autorisation attribués.



J'ai ensuite développé des diagrammes de séquence afin de pouvoir visualiser de manière beaucoup plus précise les interactions entre un usager et l'interface du site. Cela me permet de voir le cheminement des requêtes entre l'utilisateur, les formulaires ainsi que la base de données.

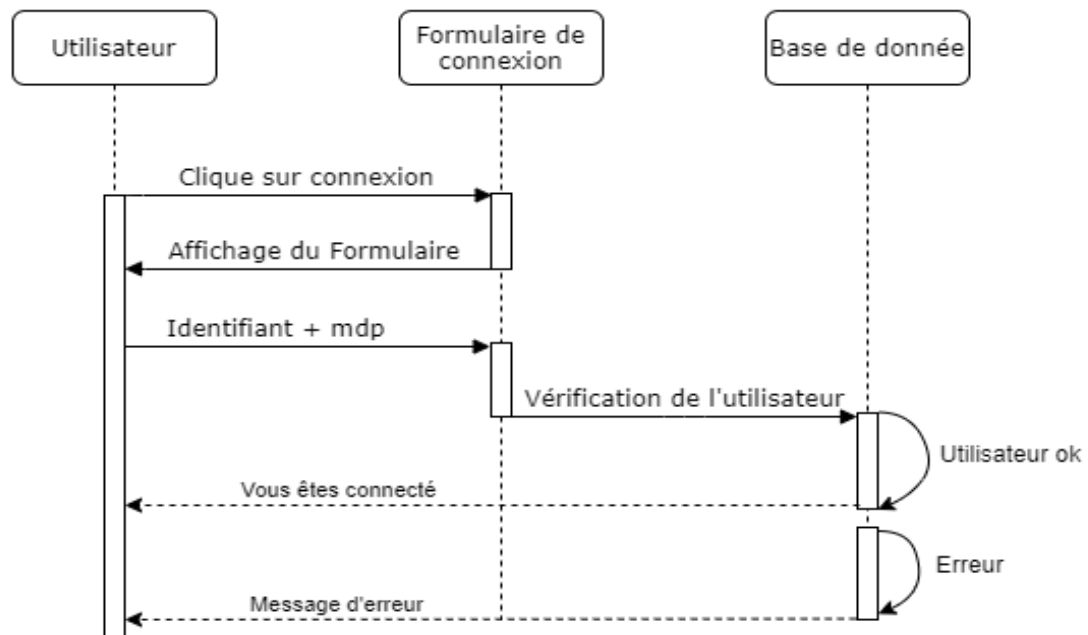
Pour ce faire j'ai détaillé trois diagrammes, à savoir l'inscription sur le site, le processus de connexion ainsi que l'envoi de messages sur le forum.

DIAGRAMME DE SEQUENCE ENREGISTREMENT DE COMPTE



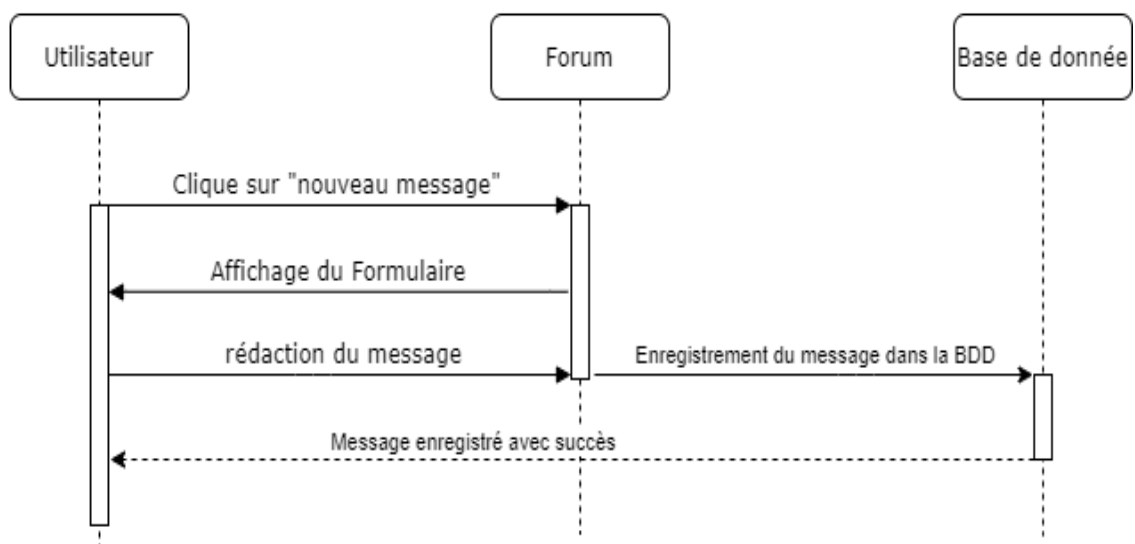
L'utilisateur doit cliquer sur le bouton de connexion afin de pouvoir afficher le formulaire d'enregistrement, il doit renseigner deux mots de passe identiques ainsi qu'une adresse email. Si les informations sont exactes, le nouvel utilisateur sera redirigé vers le formulaire de connexion afin de se connecter au site (détaillé ci-dessous).

DIAGRAMME DE SEQUENCE CONNEXION AU SITE



Afin pouvoir se connecter au site, l'utilisateur clique sur le bouton de connexion, affiche le formulaire. Si les informations envoyées sont les bonnes, il est automatiquement redirigé vers la page principale du site. Autrement, un message d'erreur sera affiché en signalant à l'utilisateur un problème concernant ses identifiants. Il y aura la possibilité de renouveler son mot de passe à l'aide d'un formulaire de renouvellement de mot de passe.

DIGRAMME DE SEQUENCE ENVOI DE MESSAGE SUR LE FORUM



Pour afficher le formulaire d'envoi de message pour le forum, j'ai souhaité mettre en place l'affichage d'une modal qui est intégrée au framework css materialize. Pour l'envoi d'un message, je me suis focalisé sur quelque chose de simple : on clique sur envoi de message, le formulaire s'affiche, on envoie le message et il s'affiche dans la page « sujet du forum concerné » quitte à rajouter des éléments durant le développement comme la modification de message ou citer un commentaire.

CREATION DU SQUELETTE ET DE LA BASE DU PROJET

Afin de démarrer mon projet j'ai commencé par construire mon squelette Symfony avec la commande :

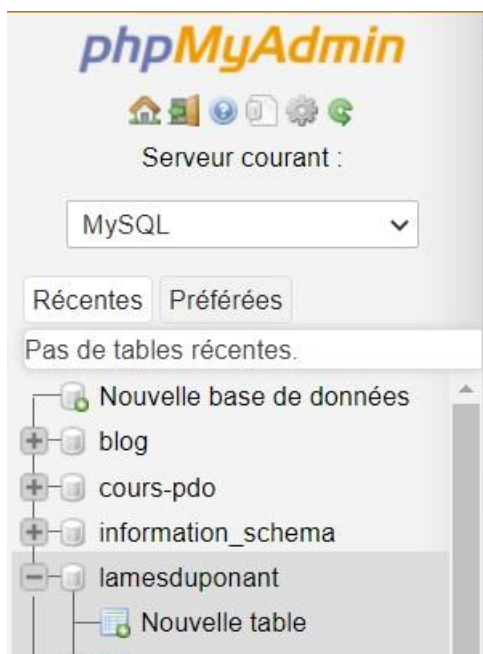
```
composer create project symfony/website skeleton leslamesduponant
```

CONSTRUCTION DE LA BASE DE DONNEE

Une fois la mise en place la base de mon projet, j'ai porté mon attention à la création de ma base de données.

Afin de pouvoir la créer correctement en ligne de commande, je suis allé modifier le fichier `.env` à la racine du projet.

Pour ce faire, j'ai modifié la variable d'environnement correspondante afin d'obtenir une relation entre mon projet et ma base de données sous MYSQL.



```
DATABASE_URL=mysql://root:"@127.0.0.1:3308/LamesDuPonant
```

Par la suite j'ai entré une ligne de commande dans le terminal :

```
Php bin/console database :create
```

Ma base de données était prête à l'emploi et automatiquement liée à mon projet.

« Dès la mise en production j'ai modifié l'accès à la base de données par les directives utilisateur/mot de passe affilié au serveur de production afin d'assurer qu'aucunes personnes malveillantes ne puisse y accéder »

CREATION D'UNE ENTITE : USER

J'ai créé l'entité **User** afin de venir lui greffer le socle de mon projet à savoir : s'enregistrer, se connecter.

La construction des entités dans Symfony est très accessible, elle se fait par le biais de ligne de commande et bénéficie d'une trame chronologique nous demandant de remplir les données à intégrer dans l'entité ainsi que le type de contenu.

Pour créer l'entité user j'ai utilisé la ligne de commande : `php bin/console make:user`

Par la suite j'ai utilisé la ligne de commande : `php bin/console make:migration`

afin de valider la migration de mon entité dans la base de données.

Ensuite j'ai validé le transfert par le biais de la commande :

`php bin/console doctrine:migrations:migrate`

CONSTRUCTION DE L'AUTHENTIFICATION

Pour créer le système d'authentification, j'ai utilisé la commande

`Php bin/console make:auth` afin de créer le formulaire de connexion avec un email ainsi qu'un mot de passe. J'ai par la suite rajouté une double vérification de mot de passe dans le formulaire d'enregistrement sur le site.

```
->add('plainPassword', RepeatedType::class, [
    'type' => PasswordType::class,
    'first_options' => [
        'attr' => [
            'class' => 'validate',
        ],
    ],
    'constraints' => [
        new NotBlank([
            'message' => 'Entrez votre mot de passe',
        ]),
        new Length([
            'min' => 6,
            'minMessage' => 'votre mot de passe doit faire minimum {{ limit }} caractères',
            'max' => 4096,
```

```

        ]),
    ],
    'label' => 'Mot de passe',
],
'second_options' => [
    'label' => 'Répéter mot de passe',
    'attr' => [
        'class' => 'validate',
    ],
],
'invalid_message' => 'le mot de passe ne correspond pas',
'mapped' => false,
]);

```

Voulant pousser au maximum l'expérience utilisateur, j'ai utilisé deux bundles afin d'apporter un plus d'ergonomie :

Pour les installer j'ai utilisé la ligne de commande

```
composer require symfonycasts/reset-password-bundle
```

```
composer require symfonycasts/verify-email-bundle
```

```
"symfonycasts/reset-password-bundle": "^1.1",
```

```
"symfonycasts/verify-email-bundle": "^1.0",
```

Le premier Bundle « reset password » créer un système de reset de mot de passe en cas de perte ou d'oubli.

Le second permet de rajouter une vérification d'email à la suite de l'inscription avec un système booléen permettant de vérifier oui ou non le compte est activé

« Pour ce dernier je n'ai pour le moment pas réussi à bloquer la connexion et de notifier d'un message en cas de non-vérification d'email, cela sera fait lors d'une prochaine mise à jour »

UTILISATEUR ET SECURITE

Pour sécuriser au mieux mon projet, je suis allé voir la documentation de Symfony en ligne afin d'avoir toutes les informations pour sécuriser mon projet. Je suis allé dans la partie **config/packages/security.yaml** afin d'y ajouter des restrictions aux accès concernant les url **/user** et **/admin** faisant en sorte que pour avoir accès à cette page, il fallait avoir le rôle ou les autorisations nécessaires.

J'ai aussi identifié les différents rôles ainsi que leurs restrictions :

- Le rôle modérateur acquiert en plus de son rôle celui d'utilisateur

- Le rôle administrateur acquiert en plus de son rôle celui de modérateur
- Le rôle super administrateur acquiert en plus de son rôle celui d'administrateur

```
# Easy way to control access for large sections of your site
# Note: Only the *first* access control that matches will be used
access_control:
  - { path: ^/admin, roles: ROLE_ADMIN }
  - { path: ^/user, roles: ROLE_USER }
role_hierarchy:
  ROLE_MODERATOR: ROLE_USER
  ROLE_ADMIN: ROLE_MODERATOR
  ROLE_SUPER_ADMIN: [ROLE_ADMIN, ROLE_ALLOWED_TO_SWITCH]
```

Afin de pouvoir sélectionner de manière plus concise les autorisations affectées à chaque utilisateur j'ai utilisé l'annotation **@IS GRANTED** utilisée dans les **controlleur** ainsi que dans les **Template** afin d'afficher ou non certaines informations, pages en fonction du rôle.

```
/**
 * @IsGranted("ROLE_USER")
 * @Route("/{id}/edit", name="user_edit", methods={"GET","POST"})
 */
public function edit(Request $request, User $user, SluggerInterface $slugger): Response
{
    $form = $this->createForm(UserType::class, $user);
    $form->handleRequest($request);

    if ($form->isSubmitted() && $form->isValid()) {
        $photoAvatarUser = $form->get('photoAvatarUser')->getData();
```

Un exemple d'annotation dans le controlleur stipulant que pour avoir accès à l'édition de son profil il fallait absolument être inscrit sur le site pour avoir le rôle User

Voici un exemple de l'utilisation de l'annotation **is_granted** dans un fichier en **html.twig** qui nous montre que pour avoir accès à la page administration dans la barre de navigation, il faut obligatoirement être connecté en tant qu'administrateur ou super administrateur afin d'afficher les liens affiliés.

```
{% if is_granted('IS_AUTHENTICATED_FULLY') %}
    {% if is_granted('ROLE_ADMIN') or is_granted('ROLE_SUPER_ADMINISTRATEUR') %}
    <li>
        <a class="waves-light waves-effect" href="{{path('admin')}}">Administration</a>
    </li>
    <li>
        <a href="{{path('contact_message_index')}}">
```

Souhaitant apporter un maximum de sécurité et éviter toutes intrusions malveillantes, j'ai étudié régulièrement les actualités sur le site de l'**OWASP** qui répertorie toutes les failles de sécurité du web ainsi que les recommandations pour s'en prémunir. J'ai lu différents articles en lien avec mon projet Symfony afin de me parer au maximum aux éventuelles « attaques ».

<https://owasp.org/www-project-top-ten/> -> liste des 10 types de risque liés à la sécurité d'application

J'ai effectué des recherches tels que « *Symfony OWASP Security* » ou « *Symfony Risk Injection* »

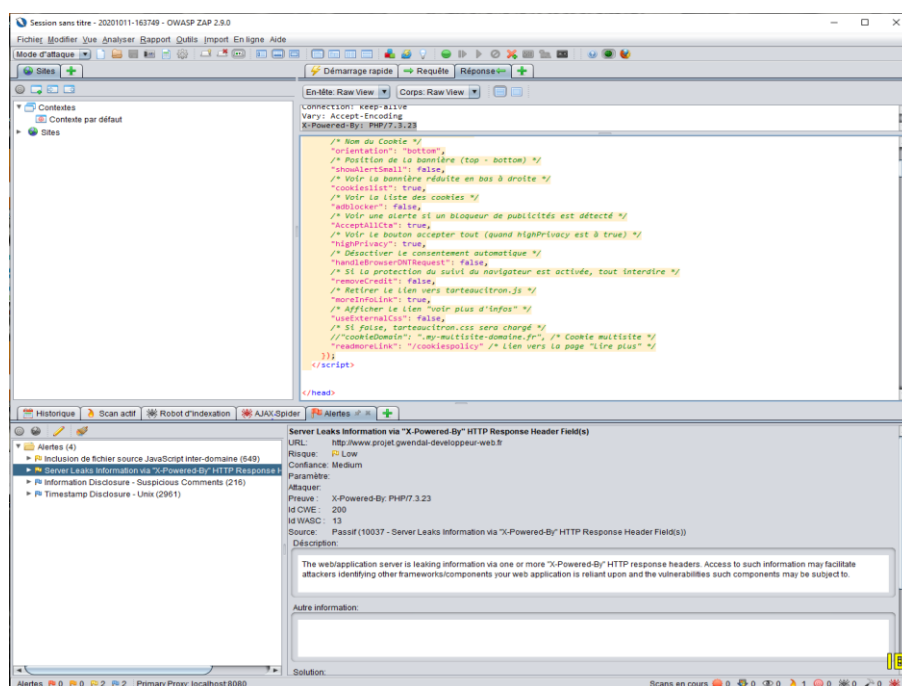
<https://nicwortel.nl/blog/2020/06/07/protect-symfony-application-against-owasp-top-10-security-risks>

nicwortel.nl est le site d'un software engineer qui a écrit un article listant tous les risques de piratage sous Symfony et comment s'en prémunir. Je me suis d'ailleurs inspiré de certains points dans le fichier **security.yaml** de Symfony afin de rajouter des éléments de sécurité auprès des rôles.

J'ai aussi pris le temps de lire la documentation de Symfony (<https://symfony.com/doc/current/security.html>), car elle est très bien expliquée et très complète pour se prémunir en cas d'attaques. <https://afsy.fr/avent/2017/16-des-astuces-de-securite-avec-symfony>

Asfy.fr (*Association Francophone Des Utilisateurs De Symfony*) est quant à lui un site francophone donnant des astuces pour mieux sécuriser son site sous Symfony avec plein d'autres conseils de codage.

Afin de tester mon site dans son intégralité, j'ai utilisé l'utilitaire Open Source **OWASP ZAP** proposé par le site OWASP qui m'a fourni des solutions afin de sécuriser au maximum mon site.



Visuel du logiciel OWASP ZAP

Le scan de l'application ne m'a donné que des résultats à un niveau de risques mineur m'informant que des informations sensibles pourrait être récupéré dans le script **tarteaucitron** et **reCAPTCHA** ainsi que dans la partie Header et Footer de la page car ils étaient directement affichés dans le code source.

Ayant scrupuleusement respecté leur installation sur mon site, je pense plutôt à un faux-risque car les tests ont été effectués en **HTTP** (le logiciel ne proposant que cette solution). Cependant une faille a été descellée au niveau d'une modal dans la partie forum, le logiciel a réussi à y rajouter du contenu sans y être connecté.

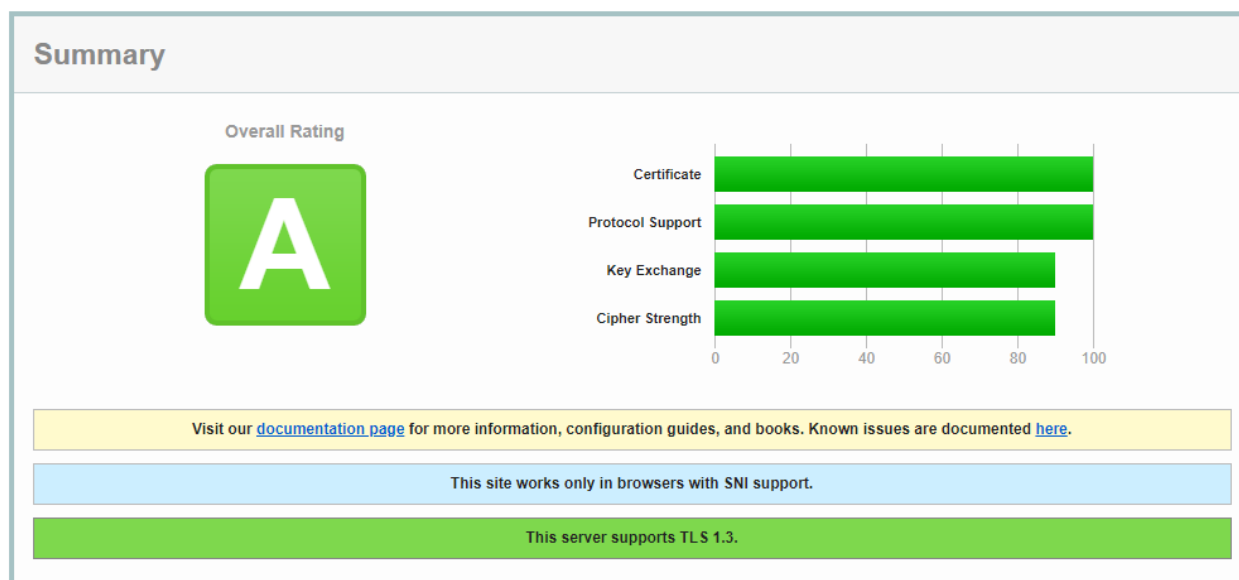
J'ai rajouté une condition **is_granted** et le problème était résolu.

```
<!-- Modal Structure -->
    {% if is_granted('IS_AUTHENTICATED_FULLY') %} → balise rajoutée demandant d'être
authenticifié pour être fonctionnelle
        <div id="modal-forum" class="formRegister z-depth-5 modal">
            {{ form_start(form) }}
            {{ form_row(form.titleSubjectForum) }}
            {{ form_row(form.contentSubjectForum) }}
            {{ form_row(form.submit) }}
            {{ form_end(form) }}
        </div>
    {% endif %} → fermeture de la balise
```

« J'ai récupéré des tutoriels traitant de la sécurité et comment s'en prémunir avec des tests personnalisés à faire sur son site. Je prendrais le temps de les mettre en pratique une fois ma formation terminée. »

SECURITE EN PRODUCTION

Une fois mon site mis en production et passé en **HTTPS** j'ai effectué d'autres tests de sécurité sur le site **Qualys** pour lequel le résultat était quasiment parfait.



Ensuite j'ai effectué un autre test sur le site **SITEGUARDING** qui vérifie la totalité des fichiers présents sur le site et détecte les malwares ainsi que les failles de sécurité éventuelles. Il effectue une vérification des liens présents sur le site en les comparant à une liste blacklistée.

General Result	GET HELP
Site	
Input: http://ipynl-general-development-works.bf	
Detailed: ipynl-general-development-works.bf	
Final URL: http://ipynl-general-development-works.bf	
IP	
195.214.167.37	
Running On	
Apache/2.4.18 Ubuntu/14.04	
Powered By	
PHP/5.6.33	
Software	
Language	
Python 2.7.9	
Version: 7.3.23	
Server	
Name: CDNetworks-Proton-Broad-VS	
Links	
js External	
https://www.google.com/recaptcha/api.js?render=ve&id=ATAMMMHqGzRNPg_uT7BU_vkZDZVQY	
https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1.min.js	
https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.2/dist/umd/popper.min.js	
https://unpkg.com/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js	
js Local	
/static/js/bundle.js	
/static/js/main.js	
Links	
/	
/login	
/register	
/list-users.html	
/about	
/contact-us.html	
/about-us.html	
/faq.html	
/privacy-policy.html	
/terms-and-conditions.html	
Website Certificate	
Cert Expires: 21 Dec 2029	
Cert Issued: Let's Encrypt	
Cert Authority: Let's Encrypt Authority X3	

Un exemple du rapport de test, je n'ai pas pu l'afficher entièrement dans mon dossier car elle faisait l'équivalent de deux pages.

Pour réaliser des tests ciblés de mon projet j'ai effectué des tests unitaires à l'aide de **PHPUnit** qui est un framework Open Source.

Une base de données de test a été créé afin de ne pas affecter celle utilisée pour mon projet. J'ai lancé plusieurs séries de tests (Repository, entités), dont l'exemple ci-dessous concerne mon entité contact pour laquelle j'ai créé une fixture que j'ai par la suite validée dans mon terminal.

Voici comment se présente une fixture :

On appelle l'entité `contactMessage`

```
namespace App\Tests;
use App\Entity\ContactMessage;
use Symfony\Bundle\FrameworkBundle\Test\KernelTestCase;
```

On utilise la classe `testContact` qui est en soi l'outil qui va nous servir à faire notre requête

```
class testContact extends KernelTestCase{
```

On crée une `public function TestTitle` (j'aurais pu lui donner n'importe quel nom)

```
public function testTitle(){
```

Appel de la variable `$contactMessage` correspondant à la classe `contactMessage` à laquelle on affine des données par le biais d'un `set` attaché au nom correspondant à ses objets `objets` (les données de l'entité)

```
$contactMessage = (new ContactMessage())
->setAuthorNameContactMessage('Salut les Amis!')
->setAuthorLastNameContactMessage('gwendal')
->setDateContactMessage(new \DateTime())
->setEmailContactMessage('gwendal@gmail.com')
->setCategoryContactMessage('Category')
->setContentContactMessage('Gwendal on test tes fixtures');
self::bootKernel();
$error = self::$container->get('validator')->validate($contactMessage);
$this->assertCount(0, $error);
}
```

On valide dans le terminal avec la ligne de commande : `php bin/console doctrine:fixtures:load --env=test`

« Le test c'est déroulé sans problème je pouvais passer à la suite de mon projet »

```
PS C:\wamp64\www\LesLamesDuPonant\LesLamesDuPonant> php bin/phpunit tests/testContact.php
PHPUnit 7.5.20 by Sebastian Bergmann and contributors.

Testing App\Tests\testContact
.
1 / 1 (100%)

Time: 446 ms, Memory: 20.00 MB

OK (1 test, 1 assertion)
PS C:\wamp64\www\LesLamesDuPonant\LesLamesDuPonant>
```

LE FORUM

Arrivant vers la fin de mon projet, j'ai souhaité créer de toutes pièces mes requêtes dans le forum afin d'en avoir une meilleure compréhension dans le but de pouvoir me débrouiller le jour où il arrive un souci dans mon code.

En premier lieu, j'ai déclaré la route `"/categories/sujet/{slug}"` ainsi que les méthodes utilisées qui seront en **Get** pour recevoir des données des sujets du forum ainsi que des messages qui lui sont associés et en méthode **Post** afin de soumettre le formulaire auprès de la base de données.

Par la suite j'ai déclaré une méthode `public function showMessage` dans laquelle je fais appel à plusieurs éléments :

\$slug : équivaut à une occurrence du sujet du forum que je souhaite récupérer (il est aussi utile pour le référencement),

request \$request : est un équivalent du `$_POST`,

Je récupère le repository de la classe **categoryForum** ainsi que de la **section** du forum afin de pouvoir les utiliser dans la pagination du site (ou breadcrumb) .

```
/**
 * @Route("/categories/sujet/{slug}", name="le_forum_subject", methods={"GET","POST"})
 */
public function showMessage(
    $slug,

    Request $request,
    SubjectForum $subjectForum,
    CategoryForumRepository $categoryForumRepository,
    HomeSectionForumRepository $homeSectionForumRepository){
```

Par la suite, nous faisons appel à Doctrine qui est un **ORM** (*Object Relational Mapping*) permettant d'effectuer des requêtes auprès de la base de données et ainsi manipuler des objets ou les modifier dans le cas échéant.

- Première partie : l’affichage

On donne une valeur à `$subjectForum` (entité) en faisant appel à doctrine qui va nous chercher le repository correspondant à la class `SubjectForum` dans lequel on utilise l’outil de recherche `findOneBy` qui permet de récupérer un ‘slug’ correspondant à la classe `SubjectForum`.

L’entité `MessageForum` ayant une relation **ManyToOne**, on récupère les messages qui en sont en relation avec le sujet intitulé `$subjectForum`. On vérifie s’il est actif et on demande de les afficher par date de publication de la plus ancienne à la plus récente avec la requête ‘DESC’.

```
$subjectForum = $this->getDoctrine()->getRepository(SubjectForum::class)->findOneBy(['slug' => $slug]);
$messageForum = $this->getDoctrine()->getRepository(MessageForum::class)->findBy([
    'subjectMessageForum'=> $subjectForum,
    'active'=> 1,
],['datePublicationMessageForum'=>'DESC']);
```

Une fois l’affichage des élément « sujets et messages » conditionnés, on va mettre en place le formulaire permettant de rajouter un message dans le forum.

- Seconde partie : préparation du formulaire

On a créé l’objet `$messageForum` qui est une instance de la classe `MessageForum`.

Ensuite on envoie une assignation par le biais de `setUser` suivi d’un `$this->getUser` afin de définir l’utilisateur lié au message. La relation **MessageUser** qui est une relation **ManyToMany** s’explique par : un message n’aura toujours qu’un utilisateur, cependant un utilisateur peut avoir plusieurs messages.

Par la suite on assigne la valeur true de type booléen en appelant la méthode `setActive()` pour activer le message et faire en sorte qu’il soit visible dans le forum.

```
$messageForum = new MessageForum();
$messageForum->setUser($this->getUser());
$messageForum->setActive(true);
$messageForum->setSubjectMessageForum($subjectForum);
```

On crée le lien relationnel entre l’entité `$messageForum` et l’entité `$subjectForum`

- Troisième partie : création du formulaire

Avec la variable `$form`, on crée un formulaire dont les différents éléments (input) ont été déclarés dans le dossier `form/messageFormType` de Symfony.

```
$form = $this->createForm(MessageFormType::class, $messageForum);
$form->add('submit', SubmitType::class, [
    'label' => 'envoyer',
    'attr' => ['class' => 'btn btn-default pull-right'],
]);
$form->handleRequest($request);
```

Je lui ai rajouté un bouton avec un label « envoyer » ainsi qu’une class CSS afin de pouvoir le modifier si besoin.

Par la suite on vérifie que le formulaire est envoyé par les méthodes « `isSubmitted` » et si le formulaire est bien valide « `isValid` »

```
if ($form->isSubmitted() && $form->isValid()) {
```

On fait appel à l'**entity manager** de Symfony afin d'inscrire les données en base de données.

Les méthodes `Persist` préparent les données à être envoyées et le `flush` inscrit les éléments du formulaire dans la base de données.

```
$entityManager = $this->getDoctrine()->getManager();

$entityManager->persist($messageForum);

$entityManager->flush();

return $this->redirectToRoute('le_forum_subject', ['slug' => $slug]);

}
```

Ensuite on retourne l'ensemble à la page courante où les éléments du formulaire ont été envoyés, la route sera '`le_forum_subject`' qui correspond à l'adresse `/sujet` suivi du slug correspondant au titre du sujet dans la barre d'adresse. Par exemple si je crée un sujet dans le forum nommé « **Bonjour à tous** », le lien complet dans la barre d'adresse sera :

<https://www.projet.gwendal-developpeur-web.fr/forum/categories/sujet/bonjour-a-tous>

- Quatrième partie : le rendu dans le Template

Cette dernière partie sert surtout à récupérer les différentes données déclarées plus haut dans notre méthode afin d'afficher le rendu.

```
return $this->render("le_forum/subjectforum.html.twig", [

    "message_forum" => $messageForum,
    'form' => $form->createView(),
    'category_forums' => $categoryForumRepository->findAll(),
    'home_section_forums' => $homeSectionForumRepository->findAll(),
    "subject_forum" => $subjectForum,

])
```

Voici une présentation d'une requête en ajax permettant d'activer ou de supprimer un élément sur le site de manière **asynchrone**. Les visuels seront dans la partie gestion de contenu.

Window.onload est l'évènement rattaché à la fenêtre active du navigateur

Let permet de déclarer la variable activer dans laquelle nous lui assignons une collection de boutons 'checkbox' présents dans le html.

Par la suite nous créons une boucle for avec une fonction 'd'écoute' du bouton checkbox en utilisant `addEventListener` qui est dans l'attente d'un « click ».

```
<script>
window.onload = () => {
  let activer = document.querySelectorAll("[type=checkbox]")
  for (let bouton of activer) {
    bouton.addEventListener("click", function () {
      let xmlhttp = new XMLHttpRequest();
      xmlhttp.open("get", `/admin/category/forum/activer/${this.dataset.id}`)
      xmlhttp.send()

    })
  }
}
```

Par la suite une requête asynchrone est envoyée au contrôleur gérant le bouton checkbox

```
/**
 * @IsGranted("ROLE_ADMIN")
 * @Route("/activer/{id}", name="activer_category_forum")
 */
public function activerCategory(CategoryForum $categoryForum)
{
    $categoryForum->setActive((
        $categoryForum->getActive()) ? false : true);
    $entityManager = $this->getDoctrine()->getManager();
    $entityManager->persist($categoryForum);
    $entityManager->flush();
    return new Response('true');
}
```

Voici la représentation de la méthode côté contrôleur.

Nous utilisons une notation ternaire qui correspond à `if else` incluant une condition, suivi d'une réponse si elle est validée.

REALISER UNE INTERFACE UTILISATEUR AVEC GESTION DE CONTENU

Afin de maximiser la gestion du site , j'ai mis en place la possibilité à l'association de mettre à jour le contenu de certaines pages. L'association aura la possibilité d'importer ses photos, de mettre du texte, le modifier et la possibilité d'activer ou non du contenu.

INTERFACE D'ADMINISTRATION

L'administration du site ne diffère que sur quelques points par rapport à l'interface utilisateur.

J'ai récupéré la barre de navigation afin de lui intégrer les différents éléments de gestion appartenant à chaque page, en prenant soin de les scinder en plusieurs blocs afin que ce soit le plus accessible possible.



GESTION DES RUBRIQUES

Quand on sélectionne une rubrique, on arrive à une page nous affichant une miniature sous forme de tableau avec différents boutons permettant d'interagir avec la rubrique.

L'administrateur a la possibilité d'activer ou désactiver un élément suivant ses besoins (expliqué dans la partie back-end **AJAX**), un bouton pour éditer la rubrique ainsi qu'un bouton pour la supprimer.

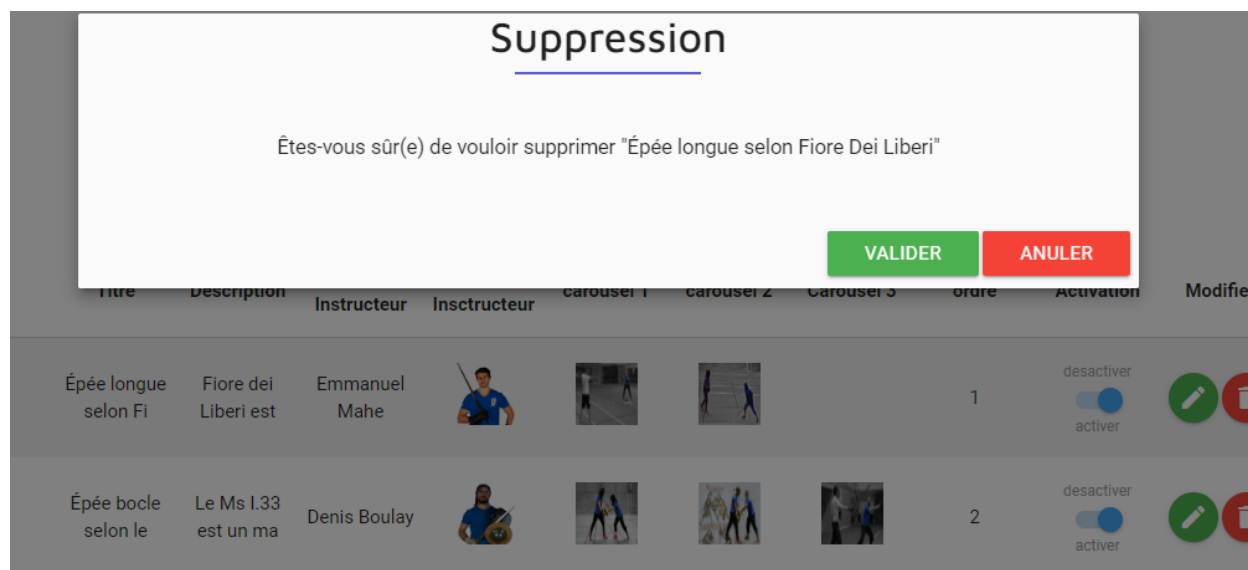
Id	Titre	Description	Nom Instructeur	Photo Instructeur	carousel 1	carousel 2	Carousel 3	ordre	Activation	Modifier
3	Épée longue selon Fi	Fiore dei Liberi est	Emmanuel Mahe					1	<div>desactiver</div> <div>activer</div>	
4	Épée boele selon le	Le Ms I.33 est un ma	Denis Boulay					2	<div>desactiver</div> <div>activer</div>	
5	Combat Viking	Le combat Viking est	Justin Deux					3	<div>desactiver</div> <div>activer</div>	

SUPPRESSION D'UNE RUBRIQUE

Lorsque l'administrateur décide de supprimer une rubrique il clique sur l'icone



Une **modal** s'affiche lui demandant s'il souhaite supprimer l'élément.



AJOUT/MAJ D'UNE RUBRIQUE

Section de la discipline affiliée
Épée Bocke

Order discipline
2

Nom de la discipline
Épée bocke selon le Ms I.33

Le Ms I.33 est un manuscrit allemand daté des années 1320, écrit en latin avec quelques termes allemands. Il enseigne une manière de combattre armé d'une épée à une main et d'une boole, un petit boudrier rond.

C'est une escrime dynamique, basée sur la maîtrise de l'épée de l'adversaire, le liage, et une défense compacte. Il en résulte un dialogue entre les adversaires, où chacun cherche à prendre l'initiative lors de la phase de contact entre les deux épées, et maîtriser l'autre.

Nom et prénom de l'instructeur
Denis Boulay

FICHER photoInstructor

FICHER imageOneDiscipline

FICHER imageTwoDiscipline

FICHER imageThreeDiscipline

MAJ LISTE

Voilà comment se présente la création/maj d'une discipline.

L'administrateur est invité à lier sa discipline en fonction de celle créée en page d'accueil avec un sélecteur.

Par la suite il devra fournir différents éléments comme du texte ainsi que des photos,


Et enfin valider avec le bouton **MAJ** situé en bas de l'écran.

Il a la possibilité de revenir à la **LISTE** des disciplines pour afficher les miniatures afin de se rendre compte s'il n'y a pas de problèmes.

MISE EN PLACE DU FRAMEWORK CSS

Le framework css materialize n'est pas nativement supporté par Symfony, pour ce faire j'ai effectué des recherches en anglais et j'ai trouvé une solution sur la plateforme GitHub m'indiquant comment intégrer materialize sous Symfony. Le **Template theme** étant adapté pour **Symfony 2**, j'ai opéré quelques modifications apportées par la communauté **Git** afin de lui assurer un support total sous Symfony 5.

« Voici un exemple de solution apportée pour la partie « input file » qui ne s'affichait pas correctement. Je me suis permis de remercier la personne pour son aide »



bernard-ng commented on 6 Jul • edited

hello, what about the input type file?

```

{% block file_widget -%}
    <{{ element|default('div') }} class="file-field input-field">
    <div class="btn">
        <span>{{ 'File' | trans }}</span>
        <input name="{{ form.vars.full_name }}"
            id="{{ form.vars.id }}"
            type="{{ form.vars.type }}"
            {{ form.vars.required ? 'required' : '' }}
        >
    </div>
    <div class="file-path-wrapper">
        <input
            class="file-path validate"
            type="text"
            {%- if attr.placeholder is defined and attr.placeholder is not none -%}
                placeholder="{{ attr.placeholder | trans }}"
            {%- endif -%}
        >
    </div>
    <{{ element|default('div') }}>
{%- endblock %}

```



toutvukantabu commented on 27 Aug

hello, what about the input type file?

```

{% block file_widget -%}
    <{{ element|default('div') }} class="file-field input-field">
    <div class="btn">
        <span>{{ 'File' | trans }}</span>
        <input name="{{ form.vars.full_name }}"
            id="{{ form.vars.id }}"
            type="{{ form.vars.type }}"
            {{ form.vars.required ? 'required' : '' }}
        >
    </div>
    <div class="file-path-wrapper">
        <input
            class="file-path validate"
            type="text"
            {%- if attr.placeholder is defined and attr.placeholder is not none -%}
                placeholder="{{ attr.placeholder | trans }}"
            {%- endif -%}
        >
    </div>
    </{{ element|default('div') }}>
{%- endblock %}

```

Thanks :)

J'ai décidé de prendre la version globale du fichier **SASS** ainsi que du fichier **Javascript** de materialize afin d'avoir une totale possibilité de modification ou d'ajouts sur mon projet ce qui s'avèrera fort utile lors de la mise en production.

Voici l'arborescence du fichier global SASS/JS fourni par materialize :

```
MyWebsite/
|--css/
| |--materialize.css <-- compiled from scss/materialize.scss
|
|--fonts/
| |--roboto/
|
|--js/
| |--materialize.js
|
|--scss/
| |--materialize.scss
| |--components/
|
|--index.html
```

Voici un exemple de mixin utilisé lors de mon projet, elle concernait un **:after** au niveau de mes titres, je lui ai rajouté la possibilité de modifier la couleur ainsi que sa longueur.

Déclaration de la mixin dans le dossier parametre.scss

```
@mixin after-title($color:#2b2bc4c5, $width:50px){

    display: block;
    height: 2px;
    width: $width;
    background-color: $color;
    content: " ";
    margin: 0 auto;
    margin-top: 10px;
}
```

Importation des **paramètres. scss** dans le fichier **materialize.scss**

```
@import "parametre";
```

Appel de la mixin avec un **@include** dans la classe

```
h1,
h2,
h4 {
    &:after {
        @include after-title($width:150px);
    }
}
```

REALISER UNE INTERFACE WEB STATIQUE ET DYNAMIQUE

Afin de rendre l'expérience utilisateur la plus interactive possible, j'ai réalisé une interface responsive adaptable sur plusieurs tailles d'écrans, types d'appareils.

J'ai utilisé les différentes balises **XL**, **L**, **M**, **S** disponibles dans le framework CSS materialize afin que les éléments s'adaptent au mieux à l'écran en fonction du ratio et de l'orientation. Leur utilisation est similaire au framework CSS **Bootstrap**, on part sur une grille de 12 colonnes que l'on peut réduire par le biais d'offset.

```
<div class="col x13 offset-x11 13 offset-11 m12 s12 z-depth-5 liste-forum">
```

```
<div class="col x18 18 m12 s12 footerOutside ">
```

Les liens de la **navbar** étant impossibles à positionner sur une largeur. J'ai utilisé un menu burger afin d'afficher une **side-nav** affichant le menu du site.



Pour activer cette dernière, je me suis référé auprès de la documentation de Materialize en ligne afin de l'activer et la modifier à ma guise.

```
//SIDENAV

$(document).ready( function () {

$('.collapsible').collapsible();

$('.dropdown-button').dropdown({
  inDuration: 300,
  outDuration: 225,
  constrainWidth: false,
  belowOrigin: true,
  stopPropagation: false
});

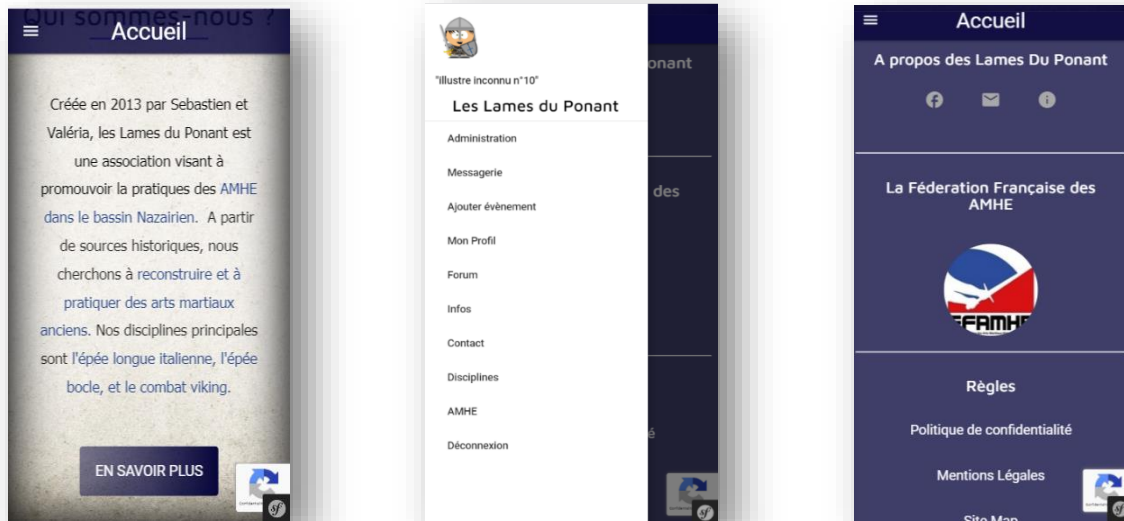
$('.button-collapse').sideNav({
  menuWidth: 300,
  edge: 'left',
  closeOnClick: false,
  draggable: true,
  onOpen: function(eL) {},
  onClose: function(eL) {

  }
});
```

Exemple : code en **javascript** utilisant la librairie **jQuery** afin d'activer la Sidenav en mode mobile.

AFFICHAGE MOBILE

Exemple de l'interface mobile : j'ai effectué quelques modifications d'affichage au niveau des icones réseaux sociaux, contact et info afin de correspondre au mieux à l'interface tactile.



Pour ce faire, j'ai utilisé des **média queries** afin d'optimiser au mieux mon site sur toutes les tailles d'écrans.

```
@media #{$small-and-down} {  
  h6 {  
    font-size: 1.4rem;  
  }  
  
  p {  
    font-size: 1.4rem;  
  }  
  
  //footer  
  
  .page-footer {  
  
    .footerLameDuPonantCenter {  
      @include border-white($border:none, $height:none);  
      border-bottom: 2px solid #ffffff;  
      border-top: 2px solid #ffffff;  
      margin-top: 20px;  
    }  
  }  
}
```

Exemple : voici une media query correspondant à l'affichage sur mobile, j'enlève les propriétés d'une mixin afin d'affilier à la classe de nouvelles règles pour afficher les « Border blanche » du Footer à l'horizontale et non à la verticale.



Pour rajouter du 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 le lieu d'entrainement de l'association.

Cette variable permet de situer la carte dans le cadre

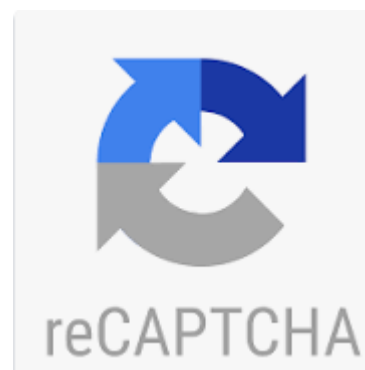
```
var mymap = L.map('mapid').setView([47.273905, -2.213766], 14);
```

Celle-ci permet d'y ajouter un marqueur personnalisé ex : la salle d'entrainement

```
var marker = L.marker([47.354449, -2.179194]).addTo(mymap);  
//Création de pop up  
marker.bindPopup("<b>Salle d'entrainement Lames Du Ponant</b>").openPopup();
```

SECURITE RECAPTCHA

Pour éviter un maximum de robots spam sur mon site, j'ai installé un outil informatique nommé **ReCAPTCHA**. C'est un script que l'on appelle sur toutes les pages du site afin de les protéger d'éventuelles attaques.



GOOGLE ANALITYCS

Afin d'analyser le trafic sur le site, j'ai installé le plugin **Google Analytics** pour avoir un récapitulatif des taux d'affluence. Le gestionnaire de cookie permet à l'utilisateur de ne pas être référencé.

MISE EN PRODUCTION



Pour la partie production du site, je suis passé par le serveur O2switch qui est une société 100% française autant pour le serveur que pour le SAV.

Le **recettage** est une phase importante du développement car elle permet d'engager une multitude de test que ce soit au niveau du support (Mobile, Tablette) et aussi faire des tests de sécurité afin de voir si son site ne comporte pas de failles (détaillé dans la partie sécurité).

Parlant de la production ci-dessus, j'ai été confronté à un sérieux problème qui m'a donné pas mal de fil à retordre (deux jours exactement). Le **carrousel** de la page d'accueil ne fonctionnait pas sur écran mobile, il était possible de le slider de gauche à droite mais impossible de cliquer sur l'image afin d'accéder à la page d'une discipline. Après une recherche avec les mots clefs « carousel click materialize don't work », j'ai enfin trouvé une réponse à ma question. La solution était écrite sur la plateforme Github et stipulait de downgrader le fichier Materialize.js dans une version antérieure pour que cela fonctionne.

Voici l'intitulé du Problème soumis par un développeur :

I have a full-width carousel using materialize with some arrows signaling next and previous carousel elements. Works great on desktop but they are not working on mobile. I have been researching it and supposedly there is a bug (no surprise) with this element that they have not covered up. I was wondering if anyone has ran into this similar problem and knew a way around it? The following code worked for one individual just not for me:

Que j'ai traduit par :

J'ai un carrousel en plein écran utilisant le framework css materialize avec des flèches indiquant suivant et précédent. Elles fonctionnent très bien en mode bureau sur un ordinateur mais ne fonctionne pas sur téléphone portable. J'ai fait des recherches et j'en suis arrivé à la conclusion qu'il y a un bug (sans surprise) avec ces éléments qui ne sont pas cachés. Je voulais demander si quelqu'un aurait eu un problème similaire et saurait comment le résoudre ? le code suivant a fonctionné juste pour une personne mais pas pour moi.

La réponse d'un membre de Github :

Remplace l'ancien code avec le code ci-dessous.

replace the above code ^ with the code given below:

```
function setupEvents() {
  if (typeof window.ontouchstart !== 'undefined') {
    view[0].addEventListener('touchstart', tap);
    view[0].addEventListener('touchmove', drag);
    view[0].addEventListener('touchend', release);
  }
  view[0].addEventListener('mousedown', tap);
  view[0].addEventListener('mousemove', drag);
  view[0].addEventListener('mouseup', release);
  view[0].addEventListener('mouseleave', release);
  view[0].addEventListener('click', click);
}
```

Voici le code que j'ai été amené à modifier afin de pouvoir faire fonctionner le click dans le carrousel sur mobile

LE REFERENCEMENT

Afin d'avoir un bon référencement sur le site, j'ai scrupuleusement respecté les standards du W3C et de Google.

J'ai effectué des tests sur le site <https://validator.w3.org> afin de corriger d'éventuels erreurs pour que mon code soit le plus propre possible.

Concernant le référencement, j'ai mis en place une méta description dans la partie Header du site et j'ai veillé à respecter l'indentation des balise H pour les titres.

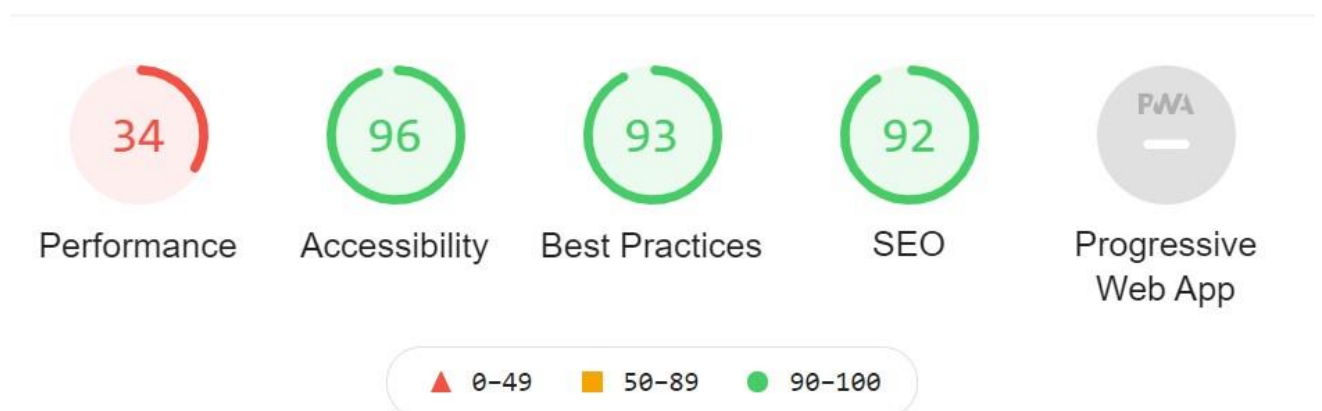
```
<meta name="description"
      content="amhe, Les Lames du Ponant association d'escrime médié-
vale à Saint Malo de Guersac, proche de Saint Nazaire">
```

Par la suite j'ai rajouté des descriptions dans chaque image pour qu'elles soient indexées par le moteur de recherche par le biais d'une balise alt

```
<div class="col xl4 l4 s12 footerLameDuPonantCenter">
  <h5>La Fédération Française des AMHE</h5>
  <a href="https://www.ffamhe.fr"></a>
</div>
```

TEST DE PERFORMANCE

J'ai effectué des tests avec le module **lighthouse**, qui m'a permis de vérifier la rapidité d'affichage de mon site en mode mobile, son accessibilité, les bonnes pratiques au niveau du code ainsi que le référencement.

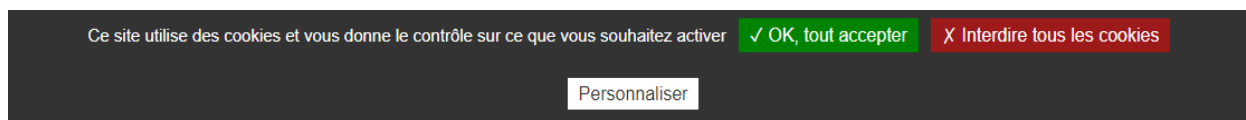


« A ce jour je travaille encore afin d'acquérir les meilleures performances possibles »

COOKIES

Pour mettre en place un système de gestion de cookies, appelé **tarteaucitron**, l'installation est assez simple : 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 src="{{ asset('js/bin//tarteaucitron.js/tarteaucitron.js') }}"></script>$
```



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

RGPD POLITIQUE DE CONFIDENTIALITE

J'ai aussi inclus une RGPD (*Règlement général sur la protection des données*) ainsi qu'une politique de confidentialité mentionnés lors de l'envoi d'un formulaire de contact par exemple.

En renseignant vos informations personnelles, vous acceptez d'être contacté(e) pour la gestion de votre compte/commande, de recevoir nos actualités et des offres promotionnelles, et vous acceptez les termes de notre [politique de confidentialité](#) Vous pouvez retirer votre consentement à tout moment directement depuis la gestion de vos données personnelles ou en nous contactant à l'adresse leslamesduponant@gmail.com

☐ Accepter La RGPD

ENVOYER ➤

L'exemple ci-dessus demande l'approbation de l'utilisateur concernant la RGPD ainsi que de la politique de confidentialité avant d'envoyer son message. J'ai rajouté un « **required true** » dans le formulaire afin qu'elle soit obligatoirement validée au préalable avant l'envoi d'un message.

L'utilisateur a aussi la possibilité de contacter l'association afin de récupérer toutes les données collectées lors de sa visite sur le site.

CONCLUSION

Dans ce rapport j'ai exposé toutes les étapes de conception de ce projet concernant la création d'un site vitrine pour l'association des Lames du Ponant permettant de mettre en avant ses activités sportives avec un environnement utilisateur dédié.

Mon projet s'est déroulé en trois grandes étapes, la première consistait à étudier les besoins de l'association afin d'établir un cahier des charges.

La seconde phase déterminait les besoins fonctionnels du site avec la création de diagrammes permettant d'élaborer l'interface utilisateur ainsi que leurs différents rôles.

La dernière concernait la conception du site en prenant en compte les aspects sécurité, responsive ainsi que les phases de tests.

Développer un projet de bout en bout fut une expérience enrichissante et maîtriser chaque étape de production m'a permis de rapidement progresser. Le principal défi était à la fois d'apprendre à coder sous Symfony et de respecter les délais du projet.

Je compte progresser en PHP et principalement en POO dans le but d'enrichir mes connaissances et d'améliorer mon niveau sous Symfony et d'autres langages dans le futur.

Aujourd'hui, l'association est satisfaite du rendu final et commence à s'approprier le site dans son intégralité. Je me suis engagé à effectuer un suivi régulier afin d'apporter certaines fonctionnalités mises de côté par manque de temps. Une refonte totale sera sûrement envisagée à l'avenir toujours sous Symfony afin de mieux factoriser mon code.