Sorenn CAVALIN

8 avenue Spinoza

20 mars 2023

Dossier Projet

Titre Professionnel Développeur Web et Web Mobile

Table des matières

[**Remerciements** 3](#_Toc124166881)

[**I.** **Introduction** 4](#_Toc124166882)

[**II.** **Résumé du Projet** 5](#_Toc124166883)

[**III.** **Cahier des Charges** 7](#_Toc124166884)

[3.1 La Conceptualisation du Web 7](#_Toc124166885)

[3.2 Utilisations et Rôles 7](#_Toc124166886)

[3.3 Mise en Place du Web 7](#_Toc124166887)

[**IV.** **Spécifications Techniques** 7](#_Toc124166888)

[4.1 Technologies du Front-End 7](#_Toc124166889)

[4.2 Technologies du Back-End 7](#_Toc124166890)

[4.3 Sécurisation du Site 7](#_Toc124166891)

[**V.** **Compétences du Référentiel Couvertes par le Projet** 7](#_Toc124166892)

[5.1 Partie Front-End 7](#_Toc124166893)

[CP1 - Maquetter une application 7](#_Toc124166894)

[CP2 – Interface statique et adaptable 7](#_Toc124166895)

[CP3 – Interface Web Dynamique 7](#_Toc124166896)

[5.2 Partie Back-End 7](#_Toc124166897)

[CP5 – Créer une Base de Données 7](#_Toc124166898)

[CP6 – Composant d’Accès aux Données 7](#_Toc124166899)

[CP7 – Partie Back-End d’une Application Web 7](#_Toc124166900)

[**VI.** **Réalisations Personnelles** 7](#_Toc124166901)

[**VII.** **Jeu d’Essai** 7](#_Toc124166902)

[**VIII.** **Veille Technologique** 7](#_Toc124166903)

[**IX.** **Problématiques** 7](#_Toc124166904)

[**X.** **Recherche Anglophone** 7](#_Toc124166905)

[**XI.** **Conclusion** 7](#_Toc124166906)

[**XII.** **Annexes** 7](#_Toc124166907)

# **Remerciements**

Je tiens tout d’abord à remercier l’équipe du PoleS et les professeurs de WebForce3pour cette merveilleuse année

Notre ? Mitra qui nous a introduit au monde des développeurs web et soutenue tout au long de cette année

La ? de notre groupe Lisa qui nous a guidée dans les démarches administratives, a fait preuve de beaucoup de compréhension et est d’une grande gentillesse

Jean-Christophe, le premier formateur de WebForce3 qui nous à appris JavaScript, est très sympathique et d’une grande pédagogie

Didier qui est arrivé après, nous à enseigner le PHP. Bien que les relations entre le formateur et les élèves étaient un peu tendues c’est un excellent professeur qui prend le temps d’expliquer plusieurs fois si nécessaire.

Numa est le dernier formateur que nous avons eu pendant cette année et, bien que son temps avec nous fut cours, il a su nous mettre en confiance avec le CMS WordPress

# **Introduction**

Je m’appelle Sorenn Cavalin.

J’ai quitté le collège a 15 ans sans diplômes pour causes de santé.

À la suite de ça, sans diplôme ni opportunité, je n’ai rien fais de notable pendant 4 ans jusqu’à ce que ma conseillère de la mission locale me propose une formation au PoleS. Hésitant, je suis allé à la journée d’information et j’ai été tenté.

J’ai pu donc intégrer la formation d’Intégrateur Développeur Web dispensée par Le PoleS Formation et Webforce3 sur la période d’avril 2022 à mars 2023.

Après ces 12 mois de formation intense, j’ai pu acquérir de nombreuses connaissances et de compétences à la fois théoriques et pratiques dans le domaine de l’intégration et du développement web, que j’ai pu en effet appliquer lors de mes deux périodes de stages et des différents projets que j’ai pu réaliser.

Tout au long de la formation, j’ai découvert plusieurs langages de programmation tels que Le HTML, CSS et JavaScript et PHP mais aussi la librairie JS JQuery et le Framework PHP Symfony.

La formation et ses expériences m’ont permis de prendre conscience du professionnel que je souhaite être et de m’investir réellement dans mon futur parcours professionnel.

Dans le cadre de la formation et en vue du passage du Titre Professionnel Développeur Web et Web Mobile j’ai réalisé un site de gestion pour une association caritative à but non-lucratif pour aider au développement de ce groupe

# **Résumé du Projet**

Dans le cadre de ma formation d’Intégrateur Développeur Web au sein du PoleS et du passage du Titre Professionnel Développeur Web et Web Mobile, j’ai réalisé un site de gestion intitulé ocrochet, pour l’association O’Crochet d’amitié.

C’est un site de gestion en ligne destiné au développement de l’association avec de la gestion de contenu notamment avec l’enregistrement des dons reçus et la création d’évènements directement sur l’application.

Le site est constitué d’une partie Front-end réalisé avec les langages HTM/CSS, JavaScript et JQuery pour le dynamisme. On y trouve dans cette partie de nombreuses fonctionnalités telles qu’un espace bénévole permettant de suivre ses dons et participations aux évènements.

Toutes les interfaces web développé sont en responsive design en utilisant les media queries.

La partie Back-end est quant à elle développée avec les langages PHP orienté objet avec l’architecture MVC et MySQL comme SGBD.

Cette partie permet de regrouper et de centraliser les fonctionnalités qui sont disponible sur le site, dont la gestion des projets, concours, dons et des comptes bénévoles.

On y gère la création, modification ou suppression de nombreuses données tel que les comptes utilisateurs ou les projets. Il y a aussi un espace don afin de pouvoir visualiser tous les dons enregistrés sur le site.

De nombreuses fonctionnalités ont été mises en place que je vous détaillerai tout au long du projet.

# **Cahier des Charges**

## La Conceptualisation du Web

Pour la conceptualisation je suis parti sur une programmation orientée objet (poo) pour une maintenance simplifiée et une architecture MVC (Modèle Vue Contrôleur) pour un dossier au rangement clair et facilité la création/recherche d’un fichier :

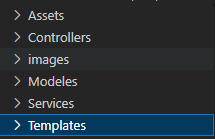


Figure : L'arborescence du dossier projet

## Utilisations et Rôles

L’application doit servir de point central de l’association d’où il est possible de gérer différents évènement (projets, concours, partenariats, etc…)

Si un utilisateur arrive sur le site et ne se connecte pas, il a accès a toutes les pages du Front-End sauf la page de profil et ne peut pas participer un concours ou remplir un don.

Quant à un utilisateur connecté (bénévole), les participations aux concours et aux dons sont possibles ainsi que le suivit de leurs actions sur le site et la modification de leur profil

Les admins ont accès à la partie Back-End du site et peuvent ajouter, modifier ou supprimer des comptes utilisateurs, dons, concours et projets.

Une image contenant texte

Description générée automatiquement

Figure : début de toutes les fonctions administrateurs

Une image contenant texte

Description générée automatiquementUne image contenant texte

Description générée automatiquement

Figure : vérifications de la connexion pour l'accès à certaines pages

## Mise en Place du Web

J’ai commencé par imaginer la base donnée et créer les entités pour me donner une idée du rendu. Une fois la bdd terminée je suis passé aux contrôleurs et la partie Back-End avec un CRUD (Create Read Update Delete) pour les entités principales, avec un formulaire et des sécurité adaptées aux entité (‘hashPassword()’ sur le mot de passe de ‘user’, vérification des dates de début et fin pour ne pas débuter après la fin, etc…). Le CRUD fini, je commence a construire le Front-End avec la page d’inscription et de connexion et je rajoute l’utilisation des rôles pour l’accès a une page ‘profil’ et l’administration du site. Pendant le développement du site j’ai pensé à relier un don a un utilisateur connecté et ai donc permis a la colonne donataire d’avoir un Integer plutôt qu’un string mais aussi a pouvoir créer un concours sans avoir a le lier a un projet, retirant l’obligation de remplir la colonne ‘projet\_id’ de la table concours.

A la base la colonne rôle contenait un tableau avec les 2 rôles pour les admins. J’ai réalisé que c’était inutile et plutôt de mettre juste ROLE\_ADMIN aux administrateurs et ne pas mettre de ‘droit’ aux user. A la place j’ai tout simplement mis une vérification de la connexion de l’utilisateur pour le rôle bénévole et la présence de rôle admin pour le Back-End.

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Figure : Fonctions ‘isAdmin’ et ‘isConnected’ pour les droits

# **Spécifications Techniques**

## Technologies du Front-End

Le Front-End a été développer en HTML, CSS et JavaScript.

La librairie de fonctions JQuery est aussi présente pour réduire la taille des fichiers JS et faciliter la manipulation du DOM et les requêtes AJAX

## Technologies du Back-End

Le Back-End est réalisé en PHP pour le serveur et MYSQL pour SGBD

## Sécurisation du Site

Une vérification des rôles s’effectue à chaque action sensible et des redirections sont lancées si un utilisateur fait une action mais ne possède pas les droits requis

Pour les entrées de données des vérifications de tous les formulaires sont effectuées avant toute les requête à la base de données

*Voir annexe n° ? pour un exemple de vérification des données*

# **Compétences du Référentiel Couvertes par le Projet**

## Partie Front-End

### CP1 - Maquetter une application

Avant le développement et la création de la base de données, une entrevue avec les responsables l’association a eu lieu pour discuter des fonctionnalités nécessaires à l’application.

J’ai reçu 3 consignes pour le développement.

* Garder l’apparence du site précédent

L’association possède déjà un site en HTML CSS fait par un bénévole. Cependant le site ne convient plus l’état actuel du groupe qui est en pleine croissance

* Pouvoir gérer le site depuis tout type d’appareil

La présidente à demander de pouvoir gérer le site depuis son téléphone ou depuis une tablette. J’ai donc inclus Bootstrap pour la partie Back-End de mon application afin de me concentrer sur le code PHP et pas sur le CSS

* Un affichage simple pour les utilisateurs

Une partie des bénévoles sont des seniors, l’affichage doit donc être intuitif pour aider la manipulation des différentes fonctionnalités du site pour les utilisateurs peu à l’aise avec la technologie

### CP2 – Interface statique et adaptable

Mon site sera adapté pour toutes les plateformes, mobile, tablette et ordinateur. Pour ce faire j’utilise les media queries. Pour cette section je vais présenter les listes de projet ou vidéos.



Figure : liste des projets sur écran > 1200px

Quand un utilisateur arrive sur cette page, une liste de 24 projets par page.

Si l’utilisateur arrive sur cette même page sur son mobile :

Une image contenant texte

Description générée automatiquement

Figure : liste de tous les projets sur mobile

Avec 3 points de rupture (1200px, 900px et 500px), j’augmente la taille des cases des div ‘projet’ pour diminuer de 1 le nombre de projets par ligne

Une image contenant texte

Description générée automatiquementUne image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquementUne image contenant texte

Description générée automatiquement

Figure : Les 4 tailles des div 'projet' pour rendre la page adaptable

Le parent de ces div projet possède les propriétés suivantes :

* Display : flex ;
* Flex-wrap : wrap ;
* Justify-content : center ;

Ces 3 propriétés permettent aux éléments enfants (div projet) de se placer automatiquement en ensemble au centre de la div et de mettre automatiquement a la ligne suivante les éléments qui dépasse du conteneur.

*Voir annexe n° ? pour un aperçu des 4 disposition de la page ‘tout les projets’*

### CP3 – Interface Web Dynamique

Pour cette section je vais présenter un objet que j’ai développé, l’objet VidéoStand.

*Voir annexe n° ? pour la classe complète*

Cet objet permet d’interagir avec les différentes vidéos présentes sur la page d’accueil de l’onglet vidéo.

*Voir annexe n° ? pour l’affichage de la page vidéo*

Dans un premier lieu je stock les 6 vidéos de la page dans la propriété ‘vidéo’ :



Je créé ensuite un getter pour récupérer uniquement la vidéo sélectionnée par l’utilisateur, la première par défaut, qui sera en avant dans la div ‘#highlight’ :

Une image contenant texte

Description générée automatiquement

Figure : une méthode qui retourne la vidéo dans l'objet JQuery qui possède l'index égal a 'videoActive'

Je pose des div ‘bloquage’ qui empêche de lancer la vidéo sur toute les div ‘case’ (les div possédant le titre de la vidéo et l’iframe qui est appelé avec le lien) et j’ajuste la taille des iframe pour ne pas dépasser du conteneur :

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Figure : Les méthodes setBloquage() et setHeight()

A l’aide de la méthode highlight je déplace la vidéo active dans la div à cet intérêt et je supprime la div de blocage sur la vidéo mise en avant :

Une image contenant texte

Description générée automatiquement

Figure : une partie de la méthode highlight

Je pose un écouteur d’évènement ‘click’ sur les div class=’case’ qui appel une fonction permettant le changement de la vidéo mise en avant

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Figure : l'écouteur d'événement et la fonction appelée

Finalement la méthode changeVideo() appelle une autre fonction (changeHighlight) qui repose une div blocage, redimensionne l’ancienne vidéo highlight et la replace dans la grille.

Une image contenant texte

Description générée automatiquement

Figure : la fonction qui permet le changement de la div 'highlight'

## Partie Back-End

### CP5 – Créer une Base de Données

La base de données a été réalisée avec la présidente de l’association pour avoir des données adaptées au besoin de l’association.

La première demande était de pouvoir avoir une idée de l’influence de l’association dans certaines régions et de pouvoir envoyer des mails aux bénévoles, j’ai donc ajouté les colonnes département (code postal), région et adresse (facultative) à la table user afin de pouvoir suivre les différentes zones de France où l’association c’est fait connaitre. Pour les mails j’ai créé une colonne ‘email’ (obligatoire) qui sert pour les mails et la connexion (la table user ne possède pas de champs pseudo). Des rôles sont aussi nécessaires pour la gestion du site. Les colonnes nom et prénom ainsi que téléphone (optionnel) sont aussi présentes pour des informations supplémentaires sur les bénévoles. Une confirmation de l’email sera aussi demandée une fois le développement de l’application terminé les colonne ‘confirmation\_token’ et active sont donc présent même si inutilisés.

La seconde fut de pouvoir garder une trace de tous les dons reçus et envoyés par l’association. La table don a alors été pensée avec deux type différents, ‘réception’ et ‘envoi’, afin de pouvoir les différencier ainsi qu’une colonne date qui se remplis seule lors de l’enregistrement pour avoir la date exacte de la création du don. Une table don\_détail avec l’identifiant du don en lien a ensuite été créée pour y entrer le contenu des dons avec nom et quantité.

Une troisième demande consiste à pouvoir créer soi-même de nouveaux projets et concours qui s’afficherons aux utilisateurs sur une page ‘projet’, d’où ils pourront y participer. Pour ce faire les concours peuvent être liés au projet avec un id dans une colonne projet\_id. Cependant, un concours peut avoir lieu sans être dans le cadre d’un projet. J’ai donc, plus tard dans mon projet, autoriser la colonne projet\_id à être vide. Pour les projets un nom ainsi que des dates de début et de fin sont nécessaires. La présidente ayant des compétences en HTML et CSS, je lui ai proposé de créer elle-même les pages projets en HTML. Après avoir reçu un avis favorable, je créé une colonne ‘page’ de type texte où sera entré la page html du projet.

Le site donne possède aussi une page vidéo qui affiche les vidéos postées sur les réseaux sociaux de l’association mais ne les héberge pas. La table vidéo contient donc une colonne lien pour y entrer le lien de la vidéo ainsi que ‘type’ pour les différents types de vidéos que l’association publie (live, tuto, etc…), un nom pour la vidéo et une colonne ‘plateforme’ qui correspond a l’hébergeur de la vidéo et sert principalement pour le trie des vidéo (comme un filtre).

Une table ‘participant’ reliant user et concours permet aux utilisateurs de s’inscrire pour les concours organisés par l’association. J’ai rajouté plus tard la colonne concours\_id à la table don pour relier les dons fait par les utilisateurs et les concours.

Pour finir j’ai encore agrandi les tables projet, concours et don avec une colonne slug pour l’affichage du Front-End et une colonne ‘supprimer’ pour ne pas supprimer les enregistrements mais juste ne plus les faires apparaitre.

Une image contenant texte, capture d’écran, intérieur

Description générée automatiquement

Figure  : Schéma de base de données

### CP6 – Composant d’Accès aux Données

Des entités (classes) Sont présente dans le dossier Modèle et correspondes à chacune des table de la base de données. Ces entités possèdent aussi des méthodes permettant d’en récupérer d’autre en lien (récupérer tous les ‘don\_detail’ depuis un don) où encore des fonctionnalités pour simplifier la manipulation des données du coté de la vue (une méthode ‘getTaille’ pour additionner toutes les colonnes quantité des ‘don\_detail’ en lien) :

Une image contenant texte

Description générée automatiquementUne image contenant texte

Description générée automatiquement

Figure : fonctions additionnelles de l'entité don

### CP7 – Partie Back-End d’une Application Web

Pour la partie Back-End de mon application je suis partie sur un affichage très simple avec une navbar qui contient un onglet vers une liste ou un formulaire de d’ajout de chaque entité ayant une page sur le Front-End. Pour ‘don\_détail’ et ‘participant’, des modifications sont possible dans les ajouts et modifications des entités don et concours :

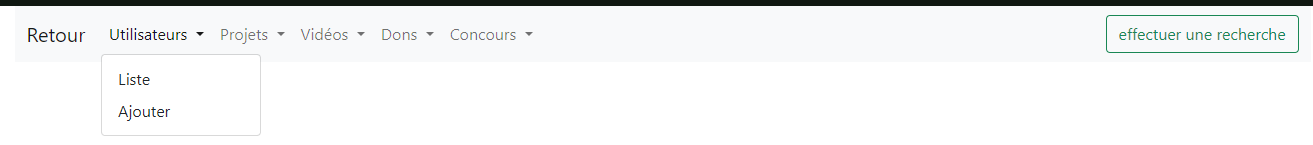


Figure : navbar admin

Pour les listes, la plupart des informations sont disponibles sur un tableau contenant chaque enregistrement de la bdd et 3 boutons, afficher, modifier et supprimer :

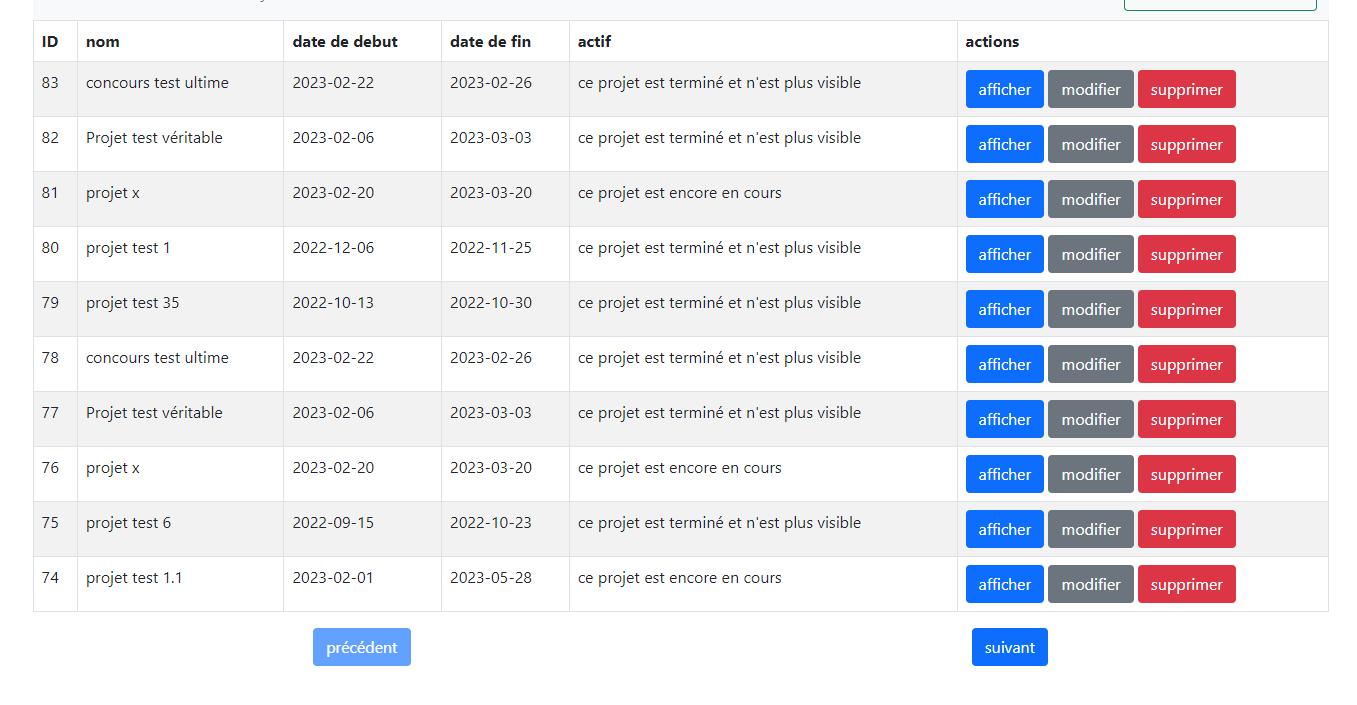


Figure : affichage des entité 'concours' en liste

La pagination est aussi gérée et sera, a l’avenir, modifiable par l’administrateur, c’est à dire modifier le nombre d’éléments par page.

Si on clique sur le bouton afficher, une fiche de l’entité en question sera affichée :

Une image contenant texte

Description générée automatiquement

Figure : fiche d'information de concours

Pour toutes les entités les informations sont affichées (sauf mot-de-passe pour user) ainsi que des données supplémentaires comme les concours en lien et la progression (actif ou finies) pour projet ou les bénévoles participant pour les concours

Don affiche aussi un tableau avec ses détails :

Une image contenant texte

Description générée automatiquement

Figure : fiche d'information d’un don avec ses détails

Lors de la suppression d’un enregistrement un pop-up rempli avec le nom de l’enregistrement ou son id apparait pour demander une confirmation et éviter les suppressions par mégarde même si la suppression n’est qu’une colonne dans la base de données :

Une image contenant texte

Description générée automatiquement

Figure : fonction JS de l'affichage pop-up supprimer avec un ajustement des données selon la ligne cliquée

Une image contenant table

Description générée automatiquement

Figure : rendu du pop-up supprimer

Si le bouton est cliqué une deuxième fois alors fois les opérations suivantes s’effectuent et un message confirmant la réussite de l’opération s’affiche alors avec un lien permettant la récupération des données pour la vue (pas effacée juste plus affichées) :

Une image contenant texte

Description générée automatiquement

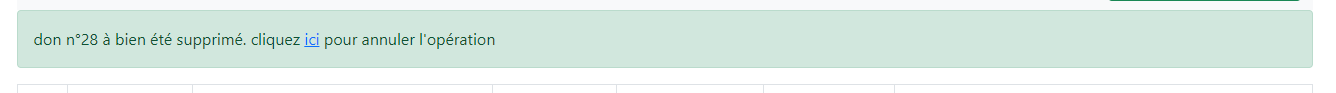


Figure : fonctions permettent la suppression d'un ou plusieurs enregistrements et son message de succès

(À l’écriture de ce dossier la fonction recover n’est pas encore développée)

Les formulaires d’ajout et de modifications sont adaptés aux entités en lien, pour cette section je vais utiliser le formulaire user :

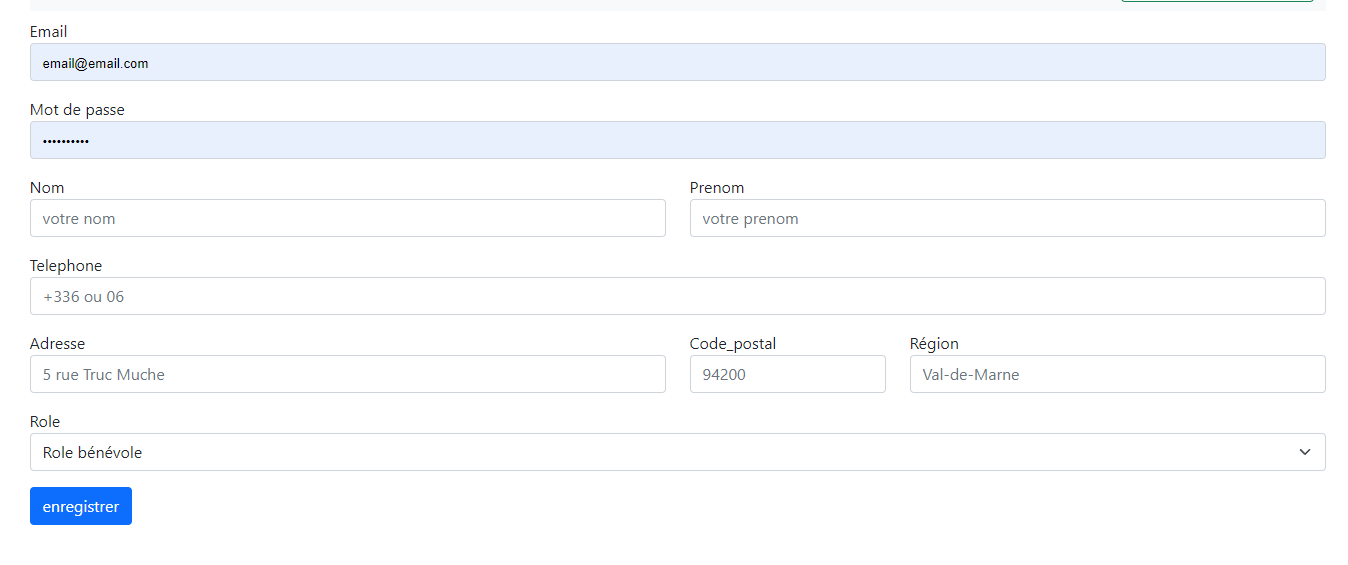


Figure : formulaire de création d’un nouveau ‘user’

Le formulaire user se remplis automatiquement lors de la modification si le champ en question est déjà rempli dans la bdd :

Une image contenant texte

Description générée automatiquement

Figure : une partie du code HTML du formulaire

Le champ téléphone doit respecter un certain paterne pour être valide le paterne est le suivant :



Signifie que la chaine de caractère peut commencer par un ‘+’, suivi de chiffre. Le tout fait entre 9 et 12 caractères

Le select rôle a une option en ‘selected’ différente selon le rôle de l’entité qui est modifiée.

Quand le formulaire est envoyé, le contrôleur appelle alors le service ‘Verificateur’ pour le traiter et effectuer les modifications/enregistrements ainsi que créer les messages d’erreur et de succès. En cas d’échec de la vérification (message d’erreur créer ou enregistrement échoué) le formulaire est réaffiché avec le ou les messages d’erreur pour guider l’utilisateur.

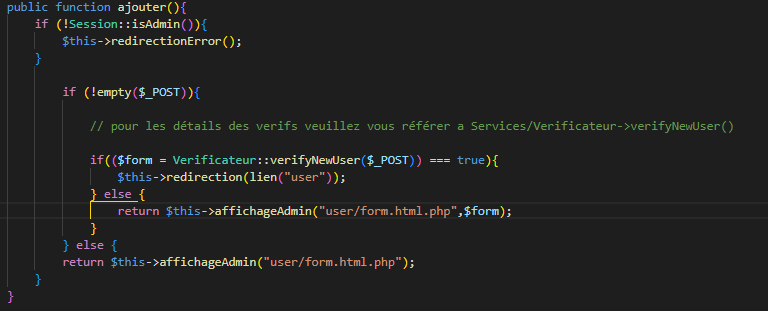


Figure : fonction ajouter du contrôleur user

# **Réalisations Personnelles**

# **Jeu d’Essai**

Pour le jeu d’essai je vais utiliser un compte administrateur et tester la création, modification et suppression d’un utilisateur.

Je commence déjà par me connecter :

Une image contenant texte

Description générée automatiquement

Figure : messages d'erreur mot de passe erroné

Dans le cas ou mon mot de passe est erroné un message d’erreur s’affiche bien et je peux donc réessayer

Une fois connecté je peux accéder à la partie Back-End de mon site en cliquant sur le logo en haut a gauche de mon site :

Une image contenant texte, nature

Description générée automatiquement

Figure : accès à la partie Back-End

Je clique ensuite sur l’onglet utilisateur puis Ajouter pour créer un nouvel utilisateur :

Une image contenant table

Description générée automatiquement

Figure : list-item-dropdown de bootstrap

J’arrive par la suite sur un formulaire avec tous les champs à remplir pour créer un compte utilisateur :

Une image contenant texte, Site web

Description générée automatiquement

Figure : formulaire d'ajout d'utilisateur

Une fois tout les champs remplis je clique sur le bouton enregistrer.

Numéro de téléphone et adresse ne sont pas obligatoires, je vais donc en remplir un sur les deux pour le test

Mon inscription est un succès et je suis redirigé sur la liste des utilisateurs avec un message confirmant la réussite de l’opération

Une image contenant texte

Description générée automatiquement

Figure : message de succès pour la création d'un nouvel utilisateur

Je vais maintenant procéder à la modification.

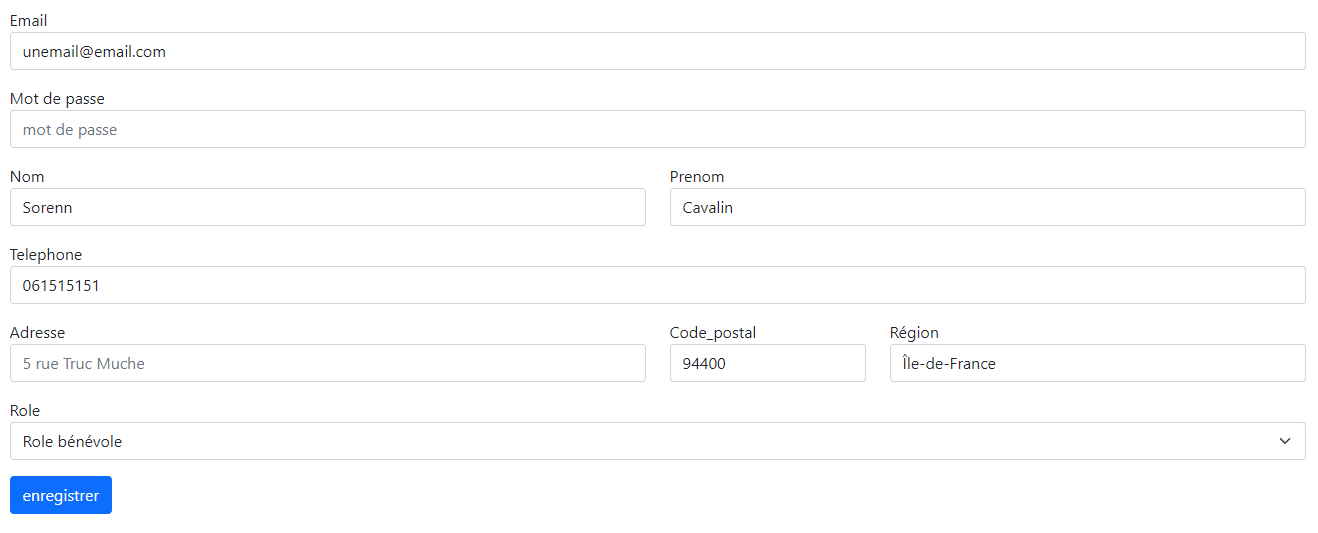


Figure : le formulaire de modification est rempli avec les données de l'entité user sélectionnée sauf pour le mot de passe

Si je clique sur enregistrer sans aucune modification, alors je suis redirigé vers la liste avec un message disant :

Une image contenant texte

Description générée automatiquement

Figure : message d'une tentative de modification sans nouvelle données

Pour la modification j’ai changé le mot de passe et ai rajouté une adresse :



Figure : message confirmant la modification d'un utilisateur

Pour la suppression j’ai décidé de placer pop-up de confirmation pour éviter les suppression par erreur ou un clic raté :

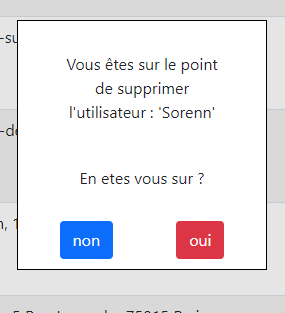


Figure : demande de confirmation pour la suppression du compte utilisateur 'Sorenn'

Le message s’adapte à la ligne où il est cliqué :

Une image contenant texte

Description générée automatiquement

Figure : code permettant la modification du message et du lien pour la suppression

# **Veille Technologique**

documentation

# **Problématiques**

J’ai rencontré des problèmes lors de la création de mon formulaire de création de don dans le Back-End de mon application.

Avec JavaScript j’ai permis la création d’un nombre Indéfini de détails ce qui permet a mon formulaire de créer a la fois un don et tout son contenu dans la table ‘don\_detail’

Une image contenant texte

Description générée automatiquement

Figure : un bouton permettant la création d'un nouveau détail

Une image contenant Site web

Description générée automatiquement

Figure : le bouton ne disparait pas une fois le détail créé

A partir de la le problème suivant est apparu : Comment récupérer et traiter un nombre inconnu de données avec PHP ?

N’ayant pas les connaissances suffisantes, je recherche sur internet : how to handle an undefined number of input php

Je trouve un résultat qui me convient sur stack overflow et j’apprend qu’il est possible de mettre des tableaux dans la propriété ‘name’ des inputs.

Je modifie alors mon code JS pour obtenir à chaque clic 2 input avec des ‘name’ contenant un array

*Voir annexe ? pour la fonction JS permettant la création des input nom et quantité*

Maintenant que mon formulaire est complet je passe donc au traitement de ces données.

Une image contenant texte, capture d’écran, écran, argent

Description générée automatiquement

Figure : traitement des détails du don

Je commence par faire une boucle sur le tableau donDetails(contient tous les détails).

J’incrémente la variable ‘boucle’ (précédemment 0) et je m’en sers pour remplir avec des index différents l’array ‘tableauDetails’ qui me servira pour l’insertion en base de données (une valeur sous la clef ‘nom’ et une sous la clef ‘qte)

Puis je crée mon don et en cas de réussite récupère depuis la base de données le dernier don(celui que je viens de créer) afin de récupérer son id et le poser dans la colonne don\_id des détails

Enfin je boucle sur mon array ‘tableauDetails’ et j’entre un par un les détails dans la base de données en les reliant a mon nouveau don avec l’id récupéré plus tôt

Une image contenant texte, écran, capture d’écran, argent

Description générée automatiquement

Figure : insertion en bdd avec une boucle

# **Recherche Anglophone**

La réponse à ma question pour une quantité indéfinie de données dans un formulaire le site que j’ai visité est stack overflow, un site anglophone. Je comprends bien l’anglais et je vais traduire avec mes mots pour trouver l’information dont j’ai besoin pour mon formulaire.

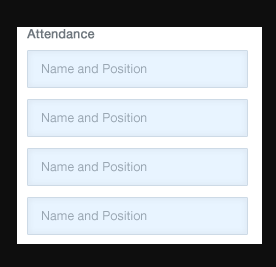
Voire annexe ? pour les captures d’écran de la question/réponse

L’intitulé du forum est :

# [**How do I parse an unknown number of inputs with PHP?**](https://stackoverflow.com/questions/9341522/how-do-i-parse-an-unknown-number-of-inputs-with-php)

*Ou : Comment traiter une quantité inconnue d’input avec php*

La personne commence par nous donner le contexte en disant qu’elle a un formulaire ressemblant à ça :



Il est aussi dit qu’un bouton pourrait rajouter d’autre champs, ce qui est très similaire a mon problème

La réponse est la suivante :

‘Just create every (similar) element of the form (in html) as an entry of an array (in php). See <http://php.net/faq.html.php#faq.html.arrays> You can just iterate over the elements then’

Ou :

‘Crée juste chaque élément similaire du formulaire en tant qu’entrée d’un tableau php (lien) tu peux ensuite faire une itération sur les éléments’

Le lien nous amène sur la page concernant les array de la documentation officielle de php.

Il y a ensuite un exemple avec la boucle en php et 4 input portant le même nom mais avec ‘[]’ a la fin pour créer un array qui aura 4 index, de 0 à 3.

# **Conclusion**

Au terme de ce travail et de cette formation intense, j’ai pu acquérir énormément de connaissances et de compétence en lien avec le métier.

Je dirais que ce projet m’a fait prendre conscience des responsabilités que l’on peut avoir lors de la réalisation d’un projet.  
Réaliser ce site de gestion était totalement un défi pour moi.

L’expérience acquise au cours de ma formation pour devenir développeur, mon application personnelle et la recherche de connaissances m’ont été bénéfiques et je pense avoir pris les bonnes décisions au cours des tâches qui m’ont été confié.

J’ai cependant perdu trop de temps sur des éléments qui n’était pas nécessaire dans l’instant et qui demander un niveau plus élevé que celui que j’avais au moment où je l’ai réalisé (je pense notamment a ma fonctionnalité de recherche qui a pris beaucoup trop de temps).

Bien évidemment l’application est loin d’être terminée et j’aurai de nombreuses occasions de tester mes capacités et d’évoluer.

Je pense avoir trouvé ma voie lors ce cette année de formation et je compte m’investir pleinement dans ce métier qu’est la développement web et web mobile

# **Annexes**

**VidéoStand**

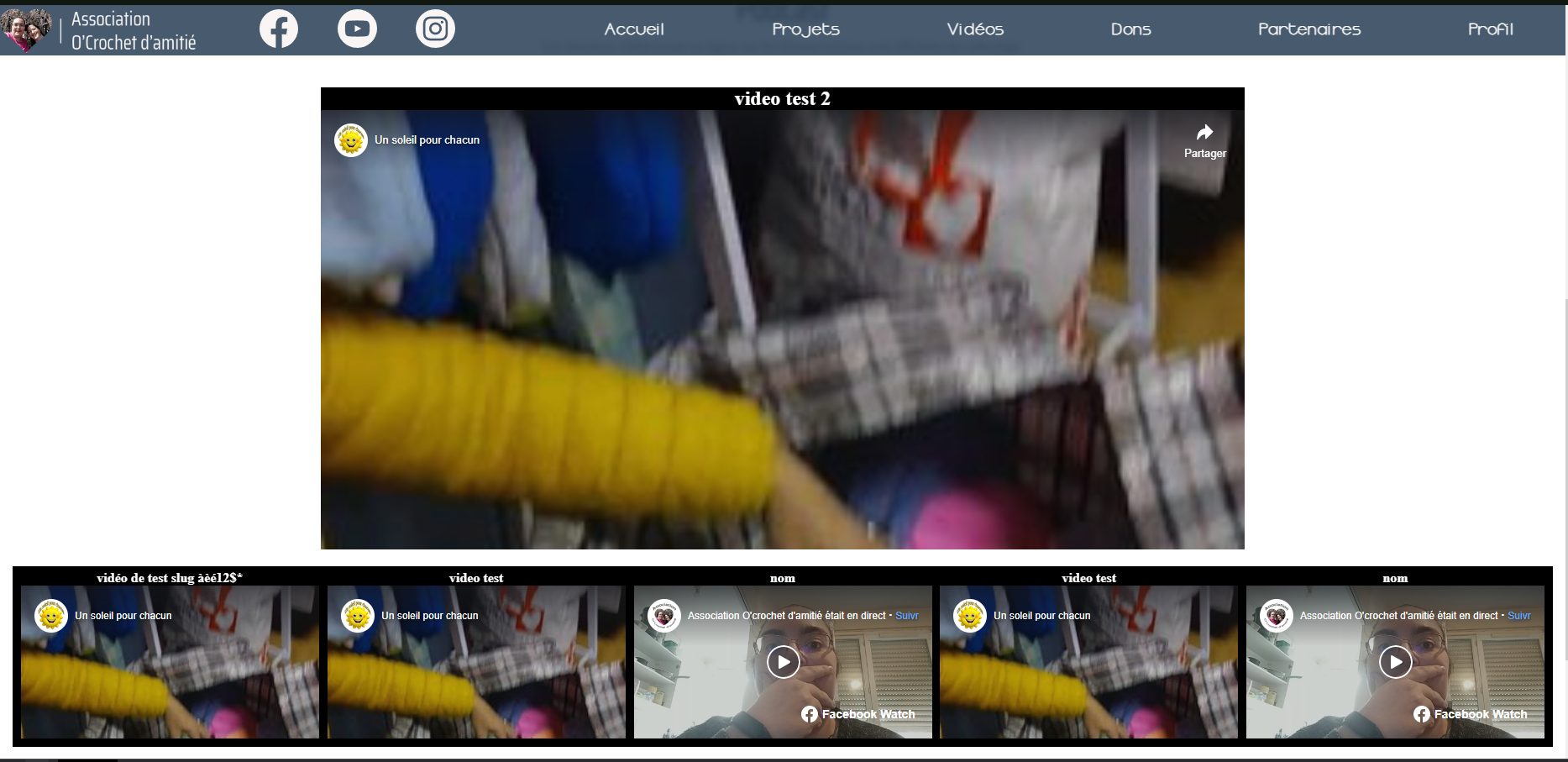


Figure : affichage de l'accueil des vidéos

Une image contenant texte, personne, capture d’écran

Description générée automatiquement

Figure : changement de la vidéo mise en avant

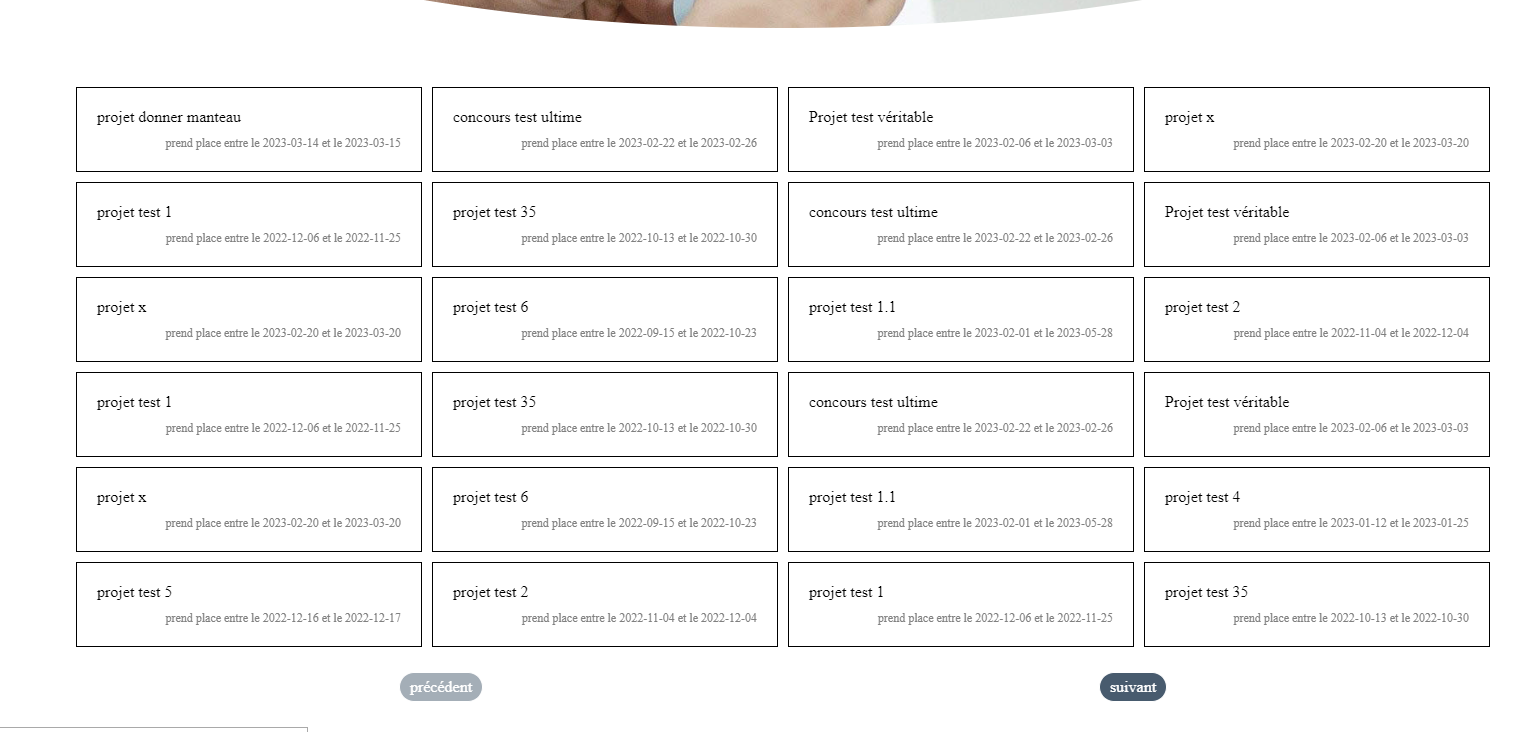


Figure : affichage de 4 projets par ligne pour un écran de plus de 1200px

Une image contenant table

Description générée automatiquement

Figure : affichage par 3 entre 1200 et 900 pixels de largeur

Une image contenant diagramme

Description générée automatiquement

Figure : en dessous de 900px l'affichage par 2

Une image contenant texte

Description générée automatiquement

Figure : un projet par ligne pour un écran sous 500px

Une image contenant texte

Description générée automatiquement

Figure : la fonction qui ajoute des inputs lors du clic sur un bouton

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Figure : La question en rapport avec ma problématique et ma section ‘recherche anglophone’