|  |  |  |  |
| --- | --- | --- | --- |
|  | ADRAR Montpellier Montpellier - Annuaire Entreprises Emploi LR |  | Simplon.co - Fabriques labellisées Grande Ecole du Numérique - In Code We  Trust #frenchtech #ess #empowerment |

|  |
| --- |
| étoile’actu |
| MEMOIRE PROFESSIONNEL |
| Titre professionnel Développeur Web & Web Mobile |

|  |
| --- |
| Pamela CALCEI  Session 2021/2022 – Adrar Pôle Numérique Montpellier |

Sommaire

1. **Introduction**
2. Remerciements3
3. Abstract4
4. Présentation5
5. Compétences à couvrir6
6. **Définir les besoins**
7. Présentation & Enoncé du besoin7
8. Contraintes techniques7
9. **Aide à la conception**
10. Cas d’utilisation9
11. Diagrammes d’activité12
12. Diagrammes de séquence15
13. **Conception**
14. Modèle conceptuel des données18
15. Modèle logique des données19
16. Base de données20
17. Maquettage24
18. Arborescence31
19. **Code**
20. Front-end32
21. Ajout d’article32
22. Modifier le profil37
23. Supprimer l’article41
24. Back-end42
25. Ajout d’article42
26. Modifier le profil5
27. Supprimer l’article5
28. **Conclusion**
29. **Introduction**
30. Remerciements

Je tiens à remercier la région Occitanie pour son financement sans lequel je n’aurais pu accéder à cette formation.

La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui je voudrais témoigner toute ma gratitude.

Je voudrais tout d’abord adresser toute ma reconnaissance à mon formateur, Monsieur Benjamin Bailly, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter ma réflexion.

Je désire aussi remercier les autres formateurs, qui m’ont fourni les outils nécessaires à la réussite de cette formation.

Je voudrais exprimer ma reconnaissance à mon entourage qui m’ont apporté leur soutien moral tout au long de ma démarche.

1. Abstract

My project is a website, which consists in sharing articles on video games. I guess you are wondering: yes, but that is pretty common?

My goal is to bring together all the features I used to inform myself about video games but in one website. So, the requirements for the website include: video games events in France and the ability to take a ticket, as well as access a gallery of these events, pictures of other people who went there previously. There are also a lot of basic features such as registration, login, etc…

Why this idea? I was thinking about an idea for my project, and I remembered that part of my life where I went to a lot of conventions, I went through so many websites to see when and where it was going on, pick up my ticket, find out how it was there and the articles where it was mentioned.

In my point of view, this website is a good combination of all that.

This website will be used by people who are passionate about video games, it will not target a particular age group.

I worked alone on this with the help of my tutor from time to time. I chose PHP oriented object, because this is the purpose of my training, with PHP, I used a database, HTML/CSS/bootstrap.

It’s not finished yet, because I was doing an internship besides and my plate was full, but some features are working.

I’m pretty proud of my project, when I started it, I didn’t know that I was able to do all these features alone. I’m not sure that I will add features but why not, I’m still thinking about it.

1. Présentation

Mon projet est un site Web, qui consiste à partager des articles sur les jeux vidéo. Je suppose que vous vous demandez : oui, mais c’est assez commun ?  
Mon objectif est de rassembler toutes les fonctionnalités que j’ai utilisées pour m’informer sur les jeux vidéo, mais dans un seul site. Ainsi, les exigences pour le site comprennent : des événements de jeux vidéo en France, la possibilité de prendre un billet, ainsi que d’accéder à une galerie de ces événements, des photos d’autres personnes qui y sont allés auparavant. Il y a aussi beaucoup de fonctionnalités de base comme l’inscription, la connexion, etc…

Pourquoi cette idée ? Je pensais à une idée pour mon projet, et je me suis souvenu de cette partie de ma vie où je suis allé à beaucoup de conventions, j’ai parcouru tellement de sites Web pour voir quand et où ça se passait, prendre mon billet, savoir comment c’était là-bas et les articles où les conventions étaient mentionnées.

À mon avis, ce site Web est une bonne combinaison de tout cela.

Ce site sera utilisé par des passionnés de jeux vidéo, il ne ciblera pas une tranche d’âge en particulier.

J’ai travaillé seule sur ce sujet avec l’aide de mon tuteur de temps en temps. J’ai choisi PHP orienté objet, parce que c’est le but de ma formation, avec PHP, j’ai utilisé une base de données(MYSQL), HTML/CSS/bootstrap.

Il n’est pas encore terminé, parce que je faisais un stage à côté et mon emploi du temps était plein, mais certaines fonctionnalités fonctionnent.

Je suis assez fière de mon projet, quand je l’ai commencé, je ne savais pas que j’étais capable de faire toutes ces fonctionnalités seule. Je ne suis pas sûr que je vais ajouter des fonctionnalités, mais pourquoi pas, je suis toujours en train d’y penser.

1. Compétences à couvrir

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **N° Fiche AT** | **Activités types** | **N° Fiche CP** | **Compétences professionnelles** | **MEM** | **DP** | |
| **1** | Développer la partie front-end d’une application web ou web mobile en intégrant les recommandations de sécurité | **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 |  |  |
| **2** | Développer la partie back-end d’une application web ou web mobile en intégrant les recommandations de sécurité | **5** | Créer une base de données |  |  |
| **6** | Développer les composants d’accès aux données |  |  |
| **7** | Développer la partie back-end d’une application web ou web mobile |  |  |
| **8** | Elaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce |  |  |

1. **Définir les besoins**
2. Présentation et énoncé du besoin

Mon projet « etoile’actu » a pour but d’informer l’utilisateur sur toutes les nouvelles sorties concernant l’esport et les jeux vidéo, en consultant des articles ou des évènements.

Le visiteur aura le droit de voir les articles mais pour accéder à la totalité il devra devenir membre donc s’inscrire.

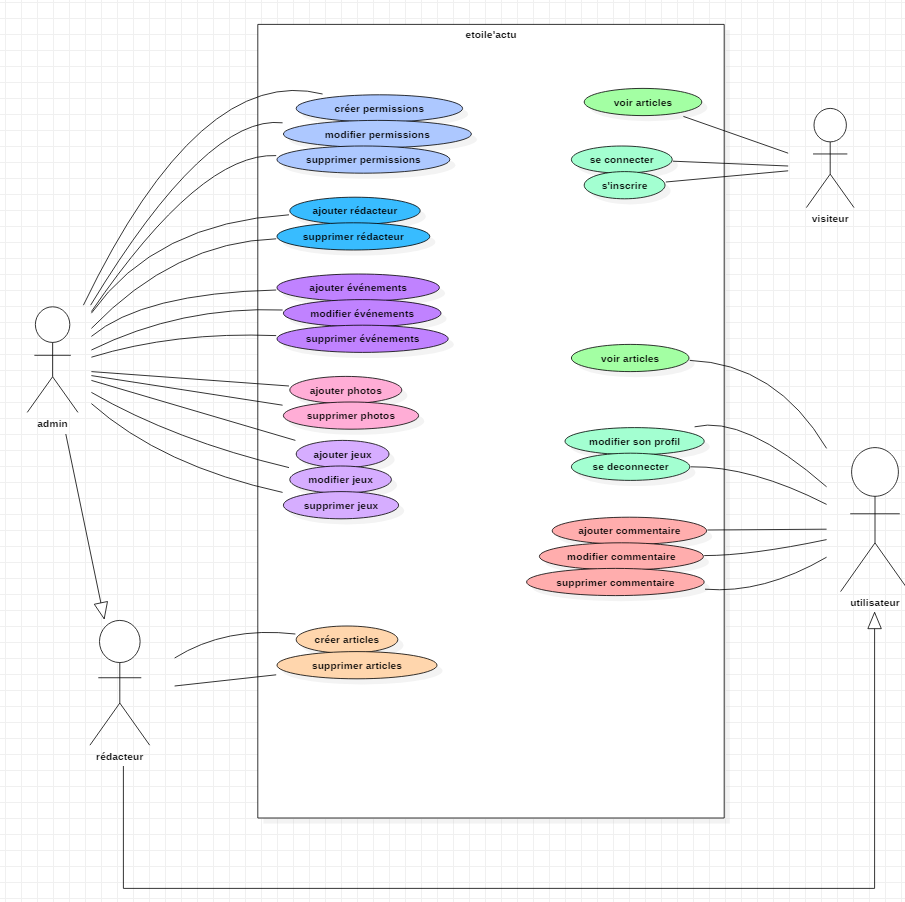
Une fois devenu membre il pourra :

* Consulter les évènements dans les différentes villes
* La possibilité d’avoir un lien direct vers la billetterie
* Voir les jeux, test sortis récemment ou non
* Pouvoir changer son profil à son image (modifier le profil)

1. Contraintes techniques

|  |  |  |
| --- | --- | --- |
| Visual Studio Code (VSCODE) à télécharger - ZDNet | J’ai tout d’abord utilisé Visual Studio Code c’est donc l’IDE que j’ai choisie pour coder mon projet |  |
| Télécharger XAMPP 8.1.2 gratuitement pour Windows/macOS/Linux | J’ai utilisé XAMPP pour héberger mon site web en local afin d’avoir un œil sur mes modifications et d’avoir accès à phpMyAdmin |  |
|  | J’ai également utilisé phpMyAdmin pour voir mes modifications dans la base de donnée (création de compte, suppression d’articles) |  |
| StarUML — Wikipédia | Pour la conception, et donc les diagrammes d’activité, de séquences et le use case (cas d’utilisation), j’ai utilisé StarUML |  |
|  | Looping m’a servi quant à lui pour le MCD et MLD |  |
|  | Pour le maquettage je me suis tournée vers AdobeXd |  |
| Qu&#39;est ce que l&#39;HTML? - Blog Kathleen B. - Ici vous trouverez la réponse ! | Au niveau des langages j’ai utilisé HyperText Markup Language 5 |  |
| Css - Transparent Css Logo Png,Css Logo Png - free transparent png images -  pngaaa.com | Pour le style, j’ai inséré du Cascading Style Sheets 3 |  |
|  | Au niveau back-end j’ai utilisé du PHP orienté objet avec le modèle MVC |  |
|  | J’ai aussi utilisé bootstrap qui est un framework CSS |  |
|  | Pour la création de base de donnée à la main, j’ai utilisé des requêtes SQL. |  |
| GitHub Logos and Usage · GitHub | Et pour finir, je me suis énormément servie de GitHub afin de pouvoir sauvegarder ce projet. |  |

1. **Aide à la conception**
2. Cas d’utilisation



Voici le cas d’utilisation, ce sont toutes les fonctionnalités auxquelles les différents acteurs peuvent accéder. Dans ce mémoire je vais vous présenter :

* L’ajout d’article qui est accessible au rédacteur ainsi qu’à l’administrateur puisque je précise que rédacteur et administrateur ne sont pas incompatibles.
* La modification de profil qui est accessible pour tous les acteurs sauf le visiteur puisqu’il n’est pas connecté.
* La suppression d’article qui est aussi accessible par le rédacteur ainsi que l’administrateur.

Dans l’entièreté du projet nous avons quatre acteurs :

* Le visiteur qui n’est pas connecté
* L’utilisateur qui l’est mais ne possède aucun droit
* Le rédacteur qui rédige les articles
* Et l’administrateur qui possède tous les droits

Voyons ensemble les différents cas d’utilisations et qui en bénéficie.

* Nous avons tout d’abord le visiteur, lui ne possède pas beaucoup de cas d’utilisation, il pourra voir les articles, se connecter ou s’inscrire
* Nous avons ensuite l’utilisateur, qui lui est connecté, il peut également voir les articles, modifier son profil, se connecter, ajouter modifier ou supprimer un commentaire
* Il y a également le rédacteur qui hérite de tous les cas d’utilisations de l’utilisateur, c’est représenté avec une relation de généralisation représenté par une flèche :

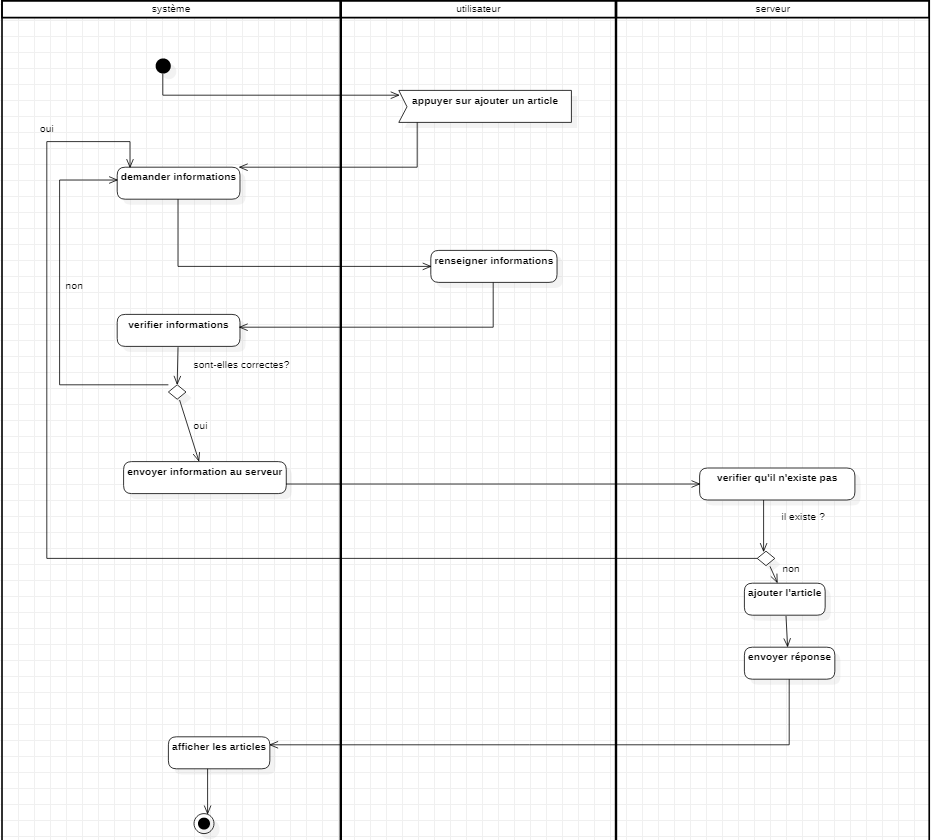


C’est-à-dire qu’il hérite de tout mais pas l’utilisateur. Il a donc les mêmes cas que l’utilisateur mais en possède d’autres dont l’utilisateur n’aura pas accès tels que : créer un article ou le supprimer.

* Pour finir, nous pouvons retrouver l’administrateur qui possède tous les droits restants et qui hérite du rédacteur qui lui hérite de l’utilisateur.

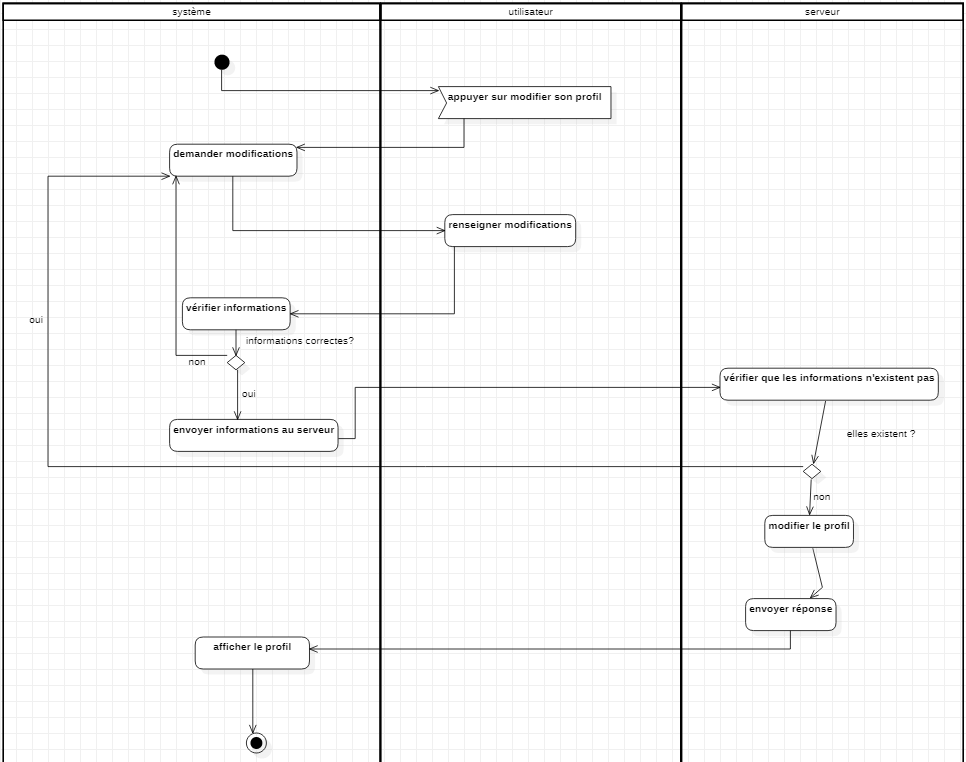
1. Diagrammes d’activité

Activité « Ajout »

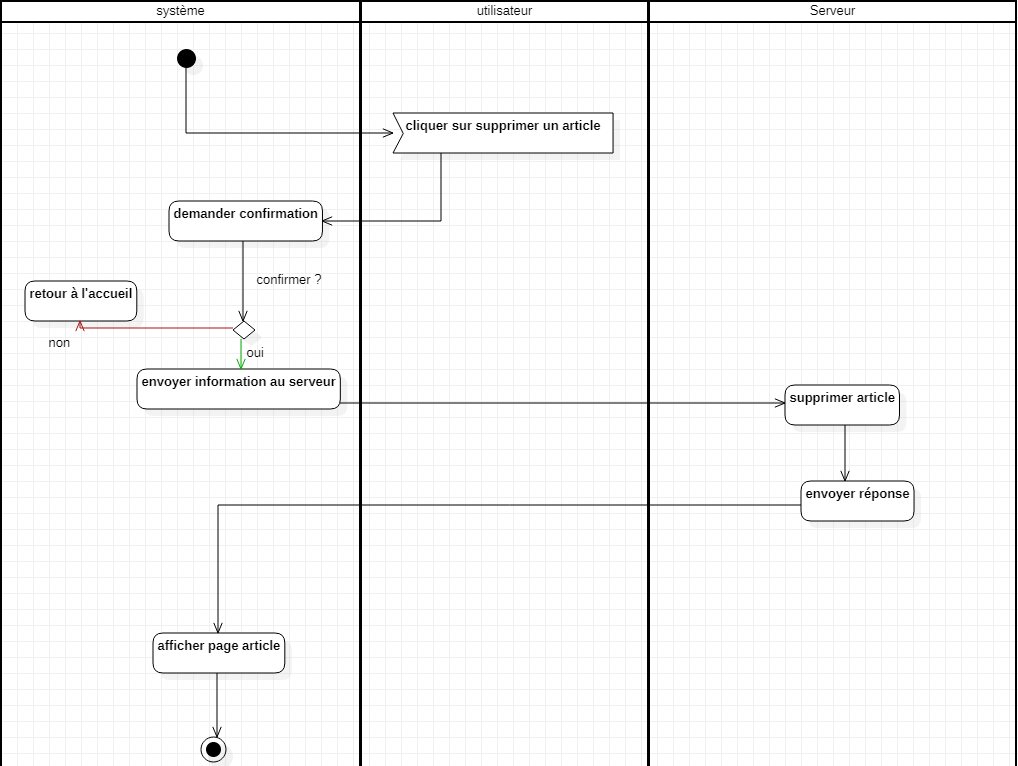


Voici le diagramme d’activité de l’ajout d’article. Le point initial provient du système, l’utilisateur, donc, en l’occurrence le rédacteur réagit en appuyant sur le bouton « ajout », le système lui demande donc les informations via un formulaire, l’utilisateur les renseigne, entre le moment où l’article apparaît et où l’utilisateur a envoyé son formulaire, plusieurs étapes seront en vigueur. Dans un premier temps, le système vérifie les informations (conditions admises par le développeur : extensions correctes ? caractères maximales non dépassées ? etc…) Si elles ne sont pas correctes, les informations seront redemandées, pour le cas inverse, elles sont envoyées au serveur (base de données). C’est ici que la base de données vérifie que ces informations n’existent pas. Si elles existent déjà, les informations seront encore une fois demandées, sinon l’article est ajouté dans celle-ci, elle envoie ensuite la réponse au système qui affiche l’article ajouté.

Activité « Modifier »



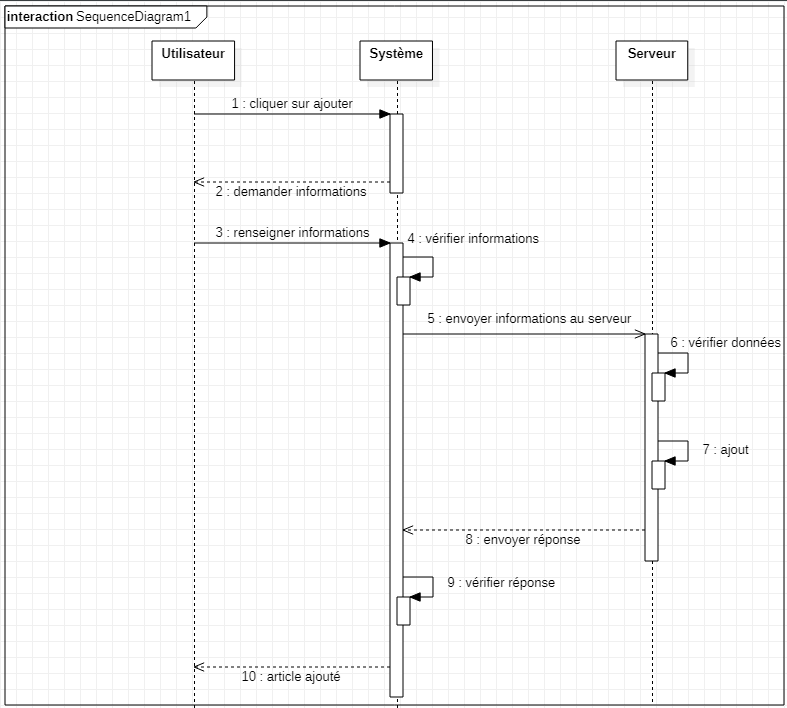
Activité « Suppression »



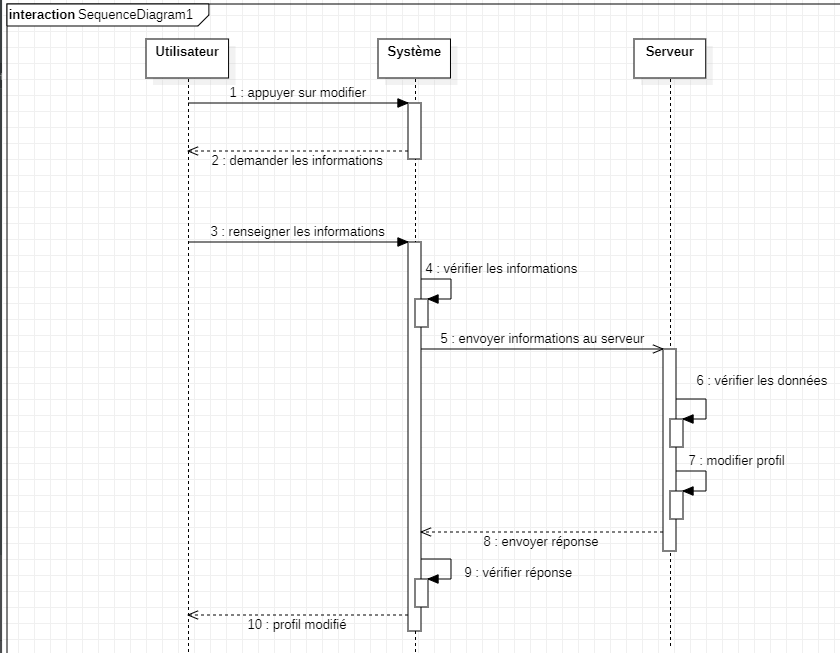
Ici, nous passons par moins de vérifications puisqu’aucune information n’est demandée, l’utilisateur clique simplement sur supprimer l’article et la base de données va gérer l’informations avant de l’envoyer au système.

1. Diagrammes de séquence

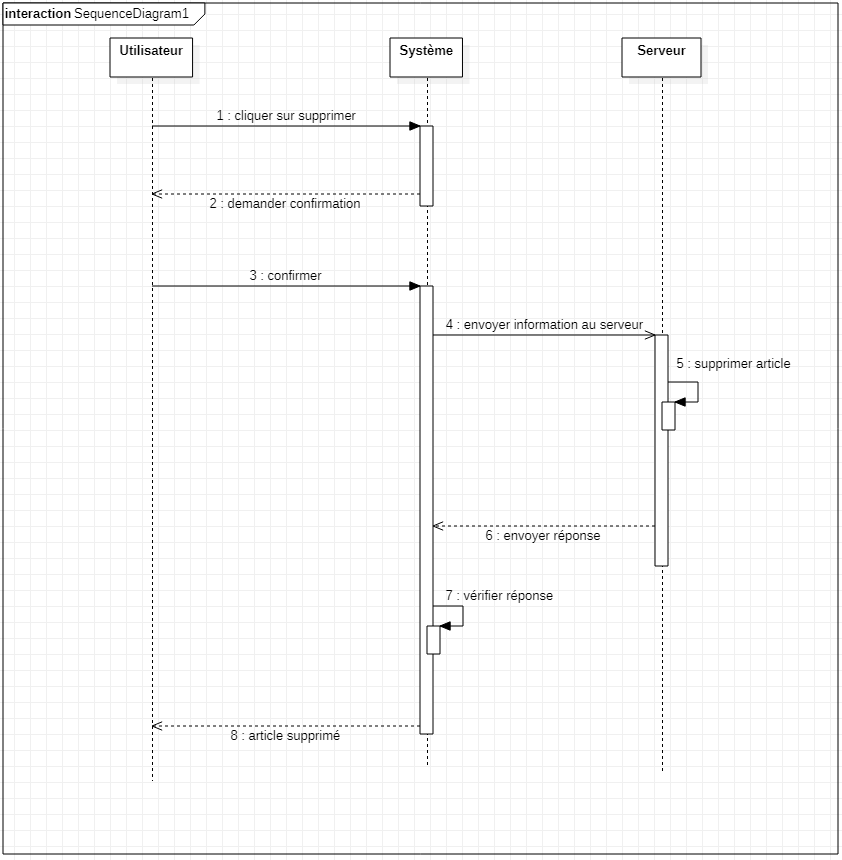
Séquence « Ajout »



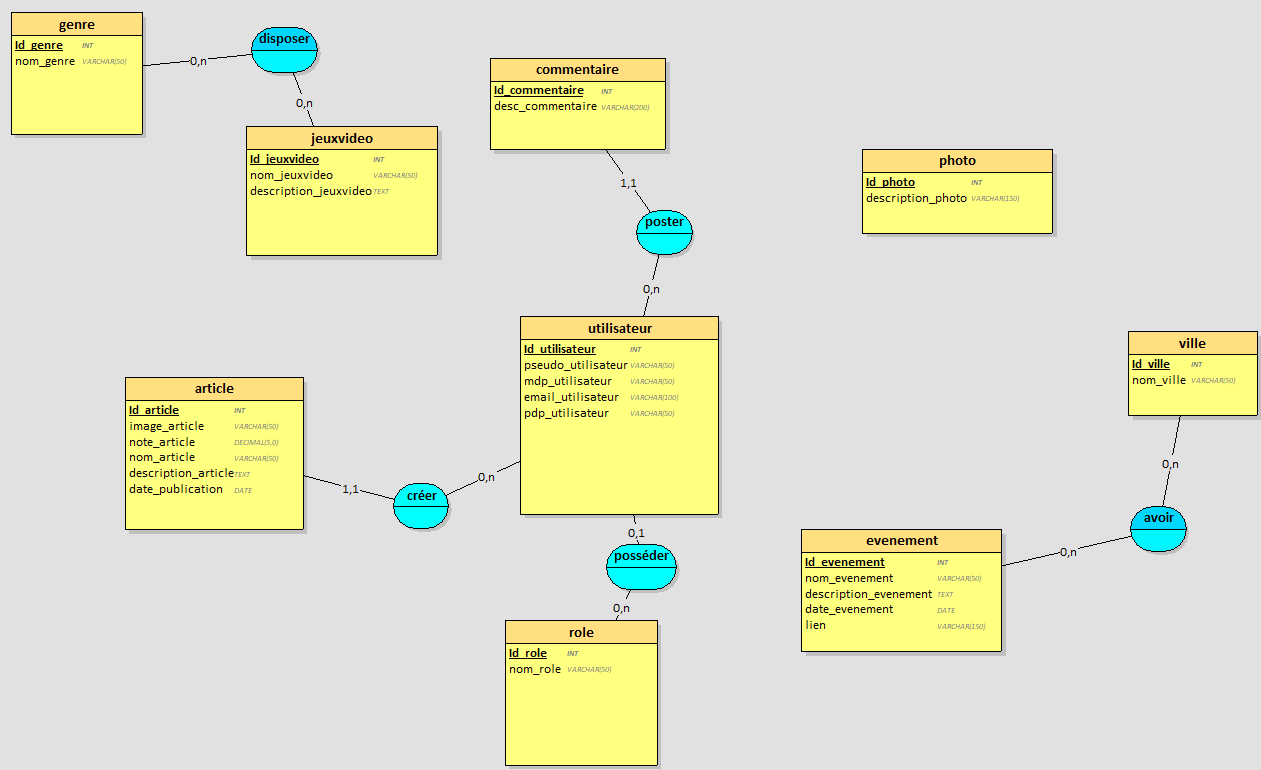
Séquence « Modifier »



Séquence « Suppression »



1. **Conception**
2. Modèle conceptuel des données



Le modèle conceptuel des données est fait d’entités et d’associations. Chaque entité possède un identifiant unique ce qui permet de la distinguer, de l’identifier. Entre chaque entité on retrouve donc une association qui possède des cardinalités qui sont définies par des nombres 0, 1 ou n (qui définit l’entièreté des nombres entiers naturels).

Prenons quelques exemples :

* L’utilisateur peut poster des commentaires d’où l’association « poster », un utilisateur peut avoir 0 ou plusieurs commentaires (0,n) mais un commentaire ne peut avoir qu’un seul et même utilisateur, et il ne peut pas de pas en avoir puisqu’il n’existerait pas (1,1).
* L’utilisateur possède un rôle, il peut avoir 0 ou 1 rôle (0,1), et le rôle peut posséder 0 ou plusieurs utilisateurs (0,n).

1. Modèle logique des données

Une image contenant texte, capture d’écran, différent, divers

Description générée automatiquement

Pour faire le modèle logique des données, nous prenons le MCD et on enlève les associations, certaines se sont transformées en clés étrangères comme :

* L’utilisateur qui possédait un rôle, on met une clé étrangère « id\_role » dans l’entité utilisateur, ce qui est représenté avec une clé étrangère qui lie deux entités.

Certaines se sont transformées en entité composée uniquement de clés étrangères comme :

* Jeux vidéo qui dispose d’un genre, c’est maintenant l’entité « disposer » avec les clés étrangères des deux entité « jeuxvidéo » et « genre ».

1. Base de donnéesUne image contenant texte

   Description générée automatiquementUne image contenant texte

   Description générée automatiquement

Ce sont les requêtes afin de créer la structure de la base de données, on y retrouve les tables, ainsi que les type de données voici celles dont j’ai eu besoin :

* INT : Nombre entier, non décimal
* AUTO\_INCREMENT : Augmente automatiquement de 1
* PRIMARY KEY design la clé primaire qui identifie chaque élément
* NOT NULL : est non nullable, c’est-à-dire qu’il est impossible que la donnée soit inexistante
* VARCHAR(..) désigne une chaîne de caractères suivie de son nombre entre parenthèses
* DATE : comme son non l’indique c’est sous forme de date
* FLOAT : nombre qui peut être décimal
* TEXT : chaîne de caractères sans limites

J’ai ensuite mis mes clés étrangères avec des ALTER TABLE afin de modifier mes tables :

Une image contenant texte

Description générée automatiquement

Nous avons donc le ALTER TABLE avec le nom de la table que nous voulons modifier, on ajoute une contrainte avec ADD CONSTRAINT on précise le nom de la FOREIGN KEY ensuite on fait REFERENCES à la table à laquelle la foreign key est référencée.

Voici le résultat :

Une image contenant table

Description générée automatiquement

Puis l’exemple d’une table :



id\_article = PRIMARY KEY

id\_util = FOREIGN KEY

1. Maquettage

Pour le logo du site web j’ai utilisé CANVA, j’ai choisi quelque chose d’assez simple qui me faisait penser à une manette de console.

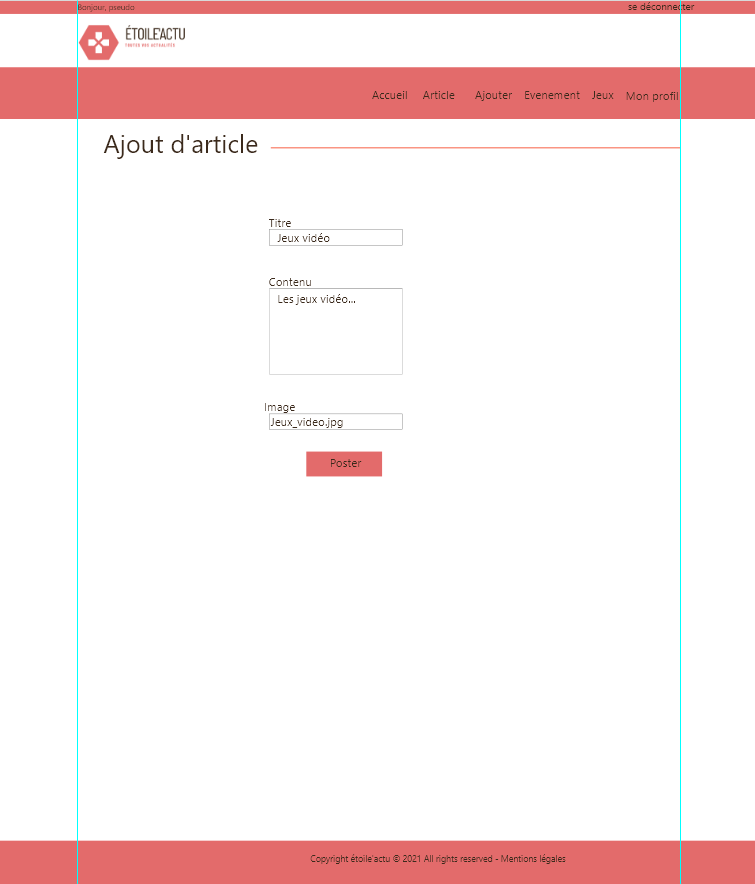


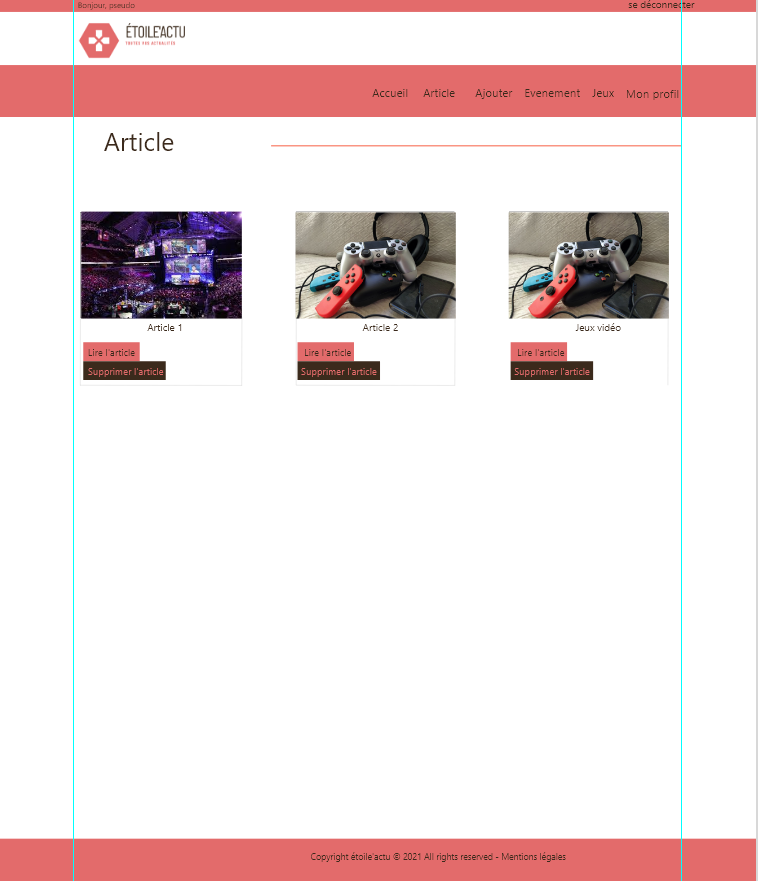
Pour les couleurs, une chose d’assez simple mais qui sautait aux yeux. Le noir était trop commun pour les écritures, j’ai donc opté pour un marron foncé.

 R :227 G : 107 B : 107

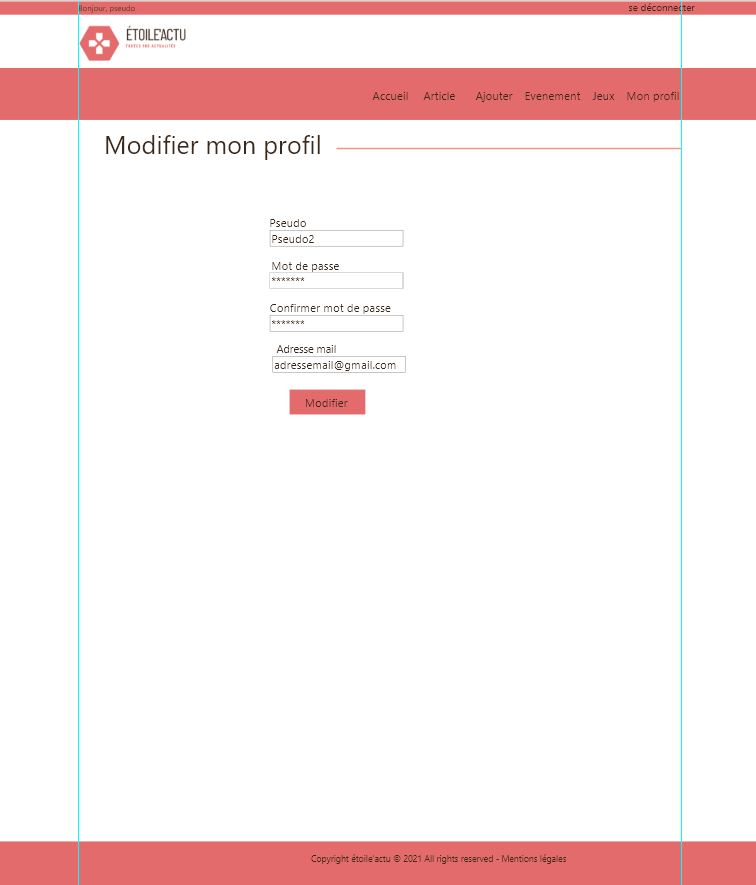
 R : 59 G : 42 B :28

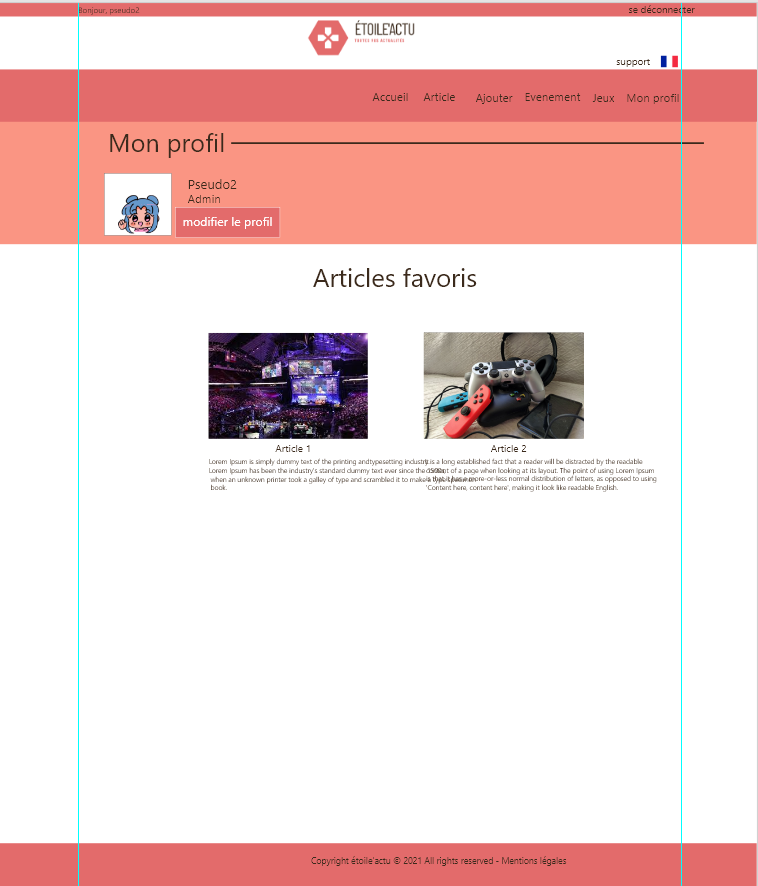
Ajout d’article



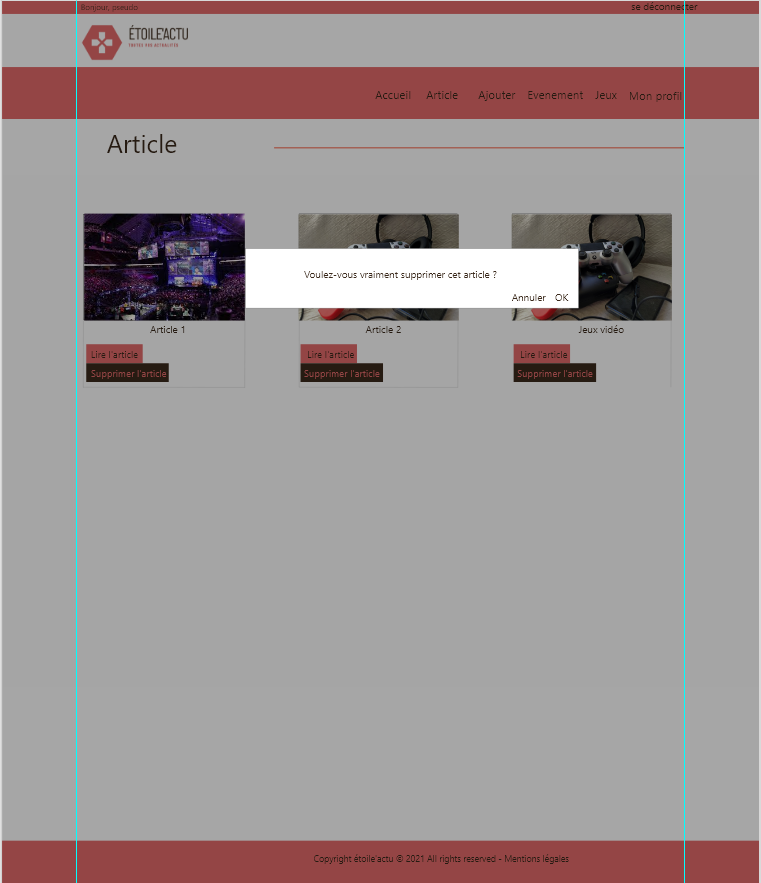


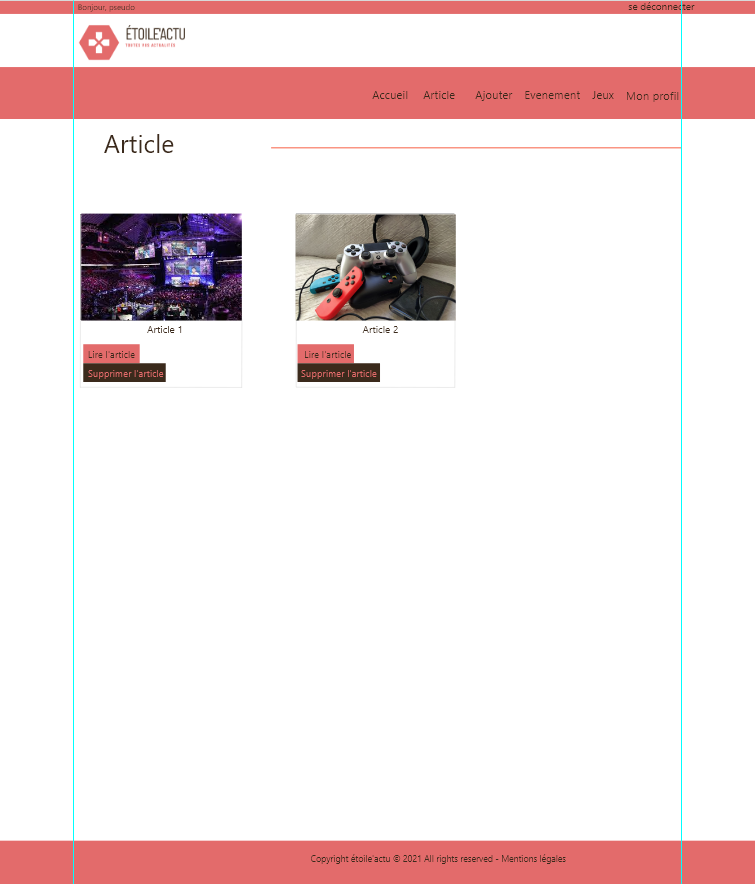
Modifier le profil





Supprimer un article





1. Arborescence

*Page d’accueil*

*Jeu 2*

*Jeu 1*

*Galerie*

*Evènement 1*

*Galerie*

*Evènement 1*

*Ville 2*

*Ville 1*

*Evènements*

*Article 2*

*Article 1*

*Article 2*

*Article 1*

*Catégorie 2*

*Catégorie 1*

*Jeux*

*Articles*

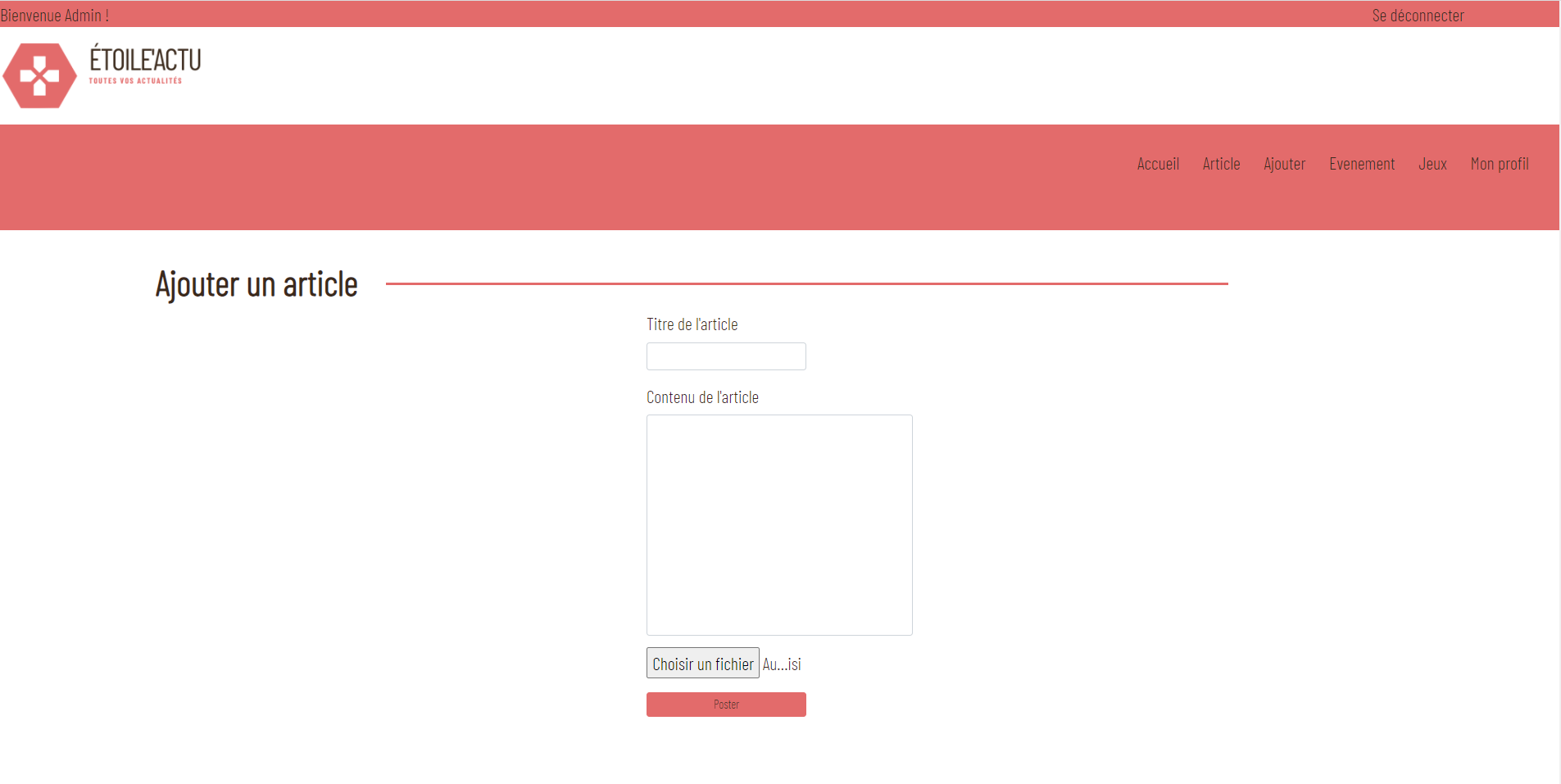
1. **Code**
2. Front-end
3. Ajout d’article



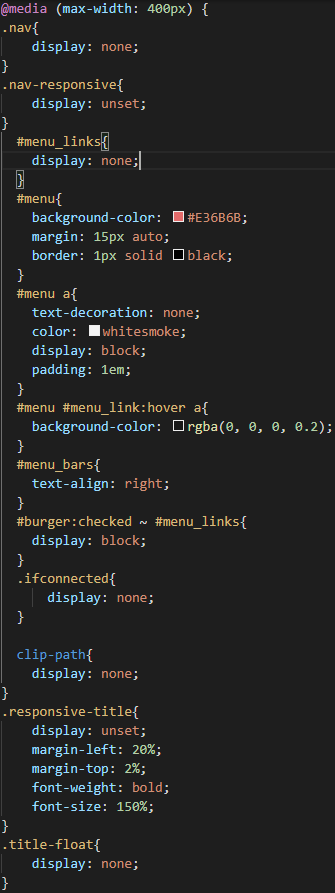


J’ai utilisé un formulaire simple en oubliant pas de mentionner chaque input pour pouvoir les récupérer sous forme de $\_POST

Résultat :



Responsive



Résultat





1. Modifier le profil

Une image contenant texte

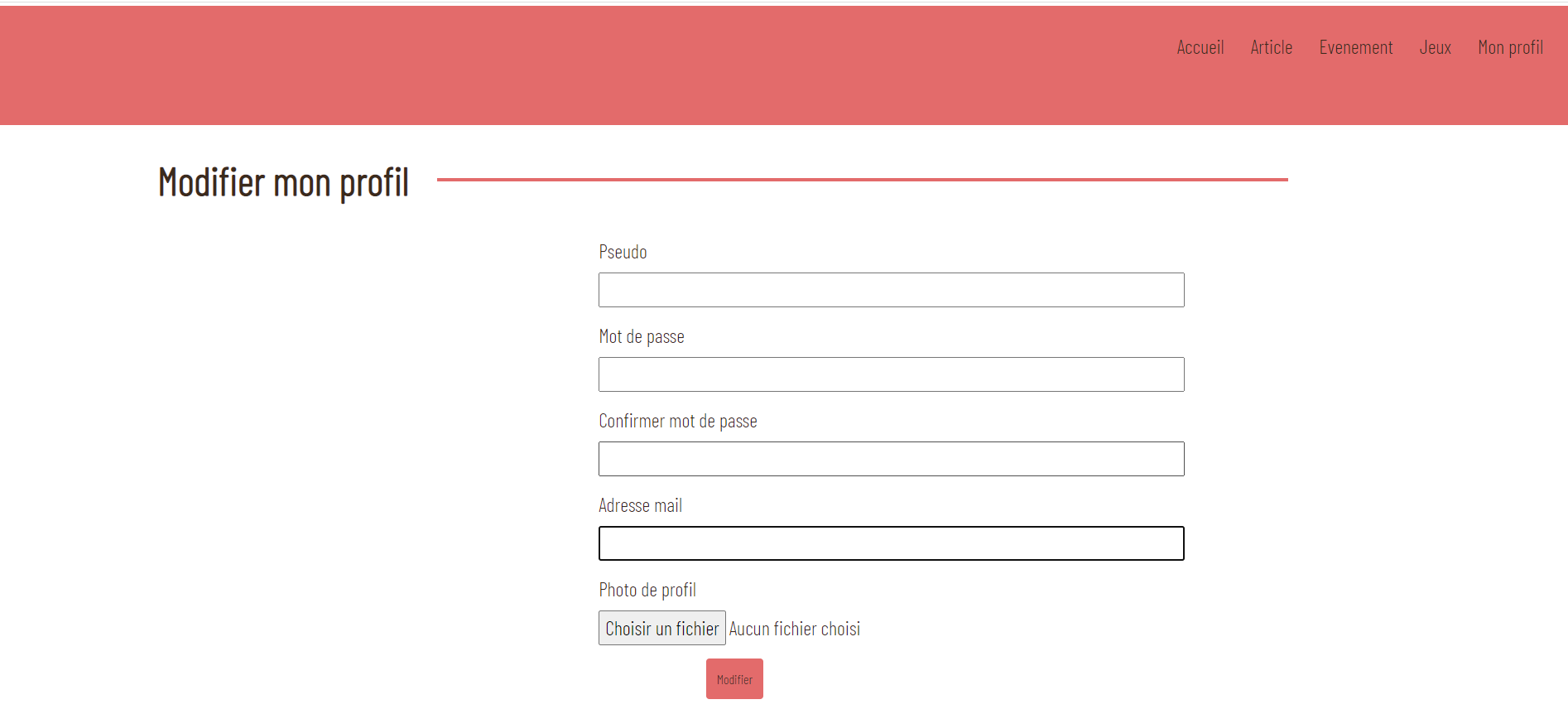
Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Encore une fois , formulaire assez simple en nommant toujours les input.

Résultat :

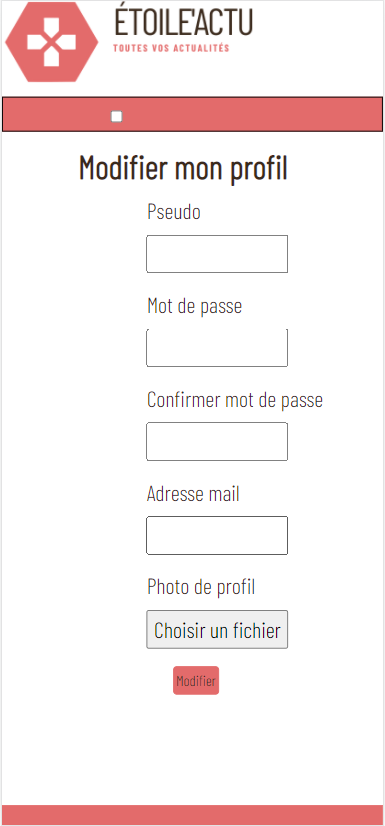


Responsive

Une image contenant texte

Description générée automatiquement

Résultat



C. Supprimer l’article

Une image contenant texte

Description générée automatiquement

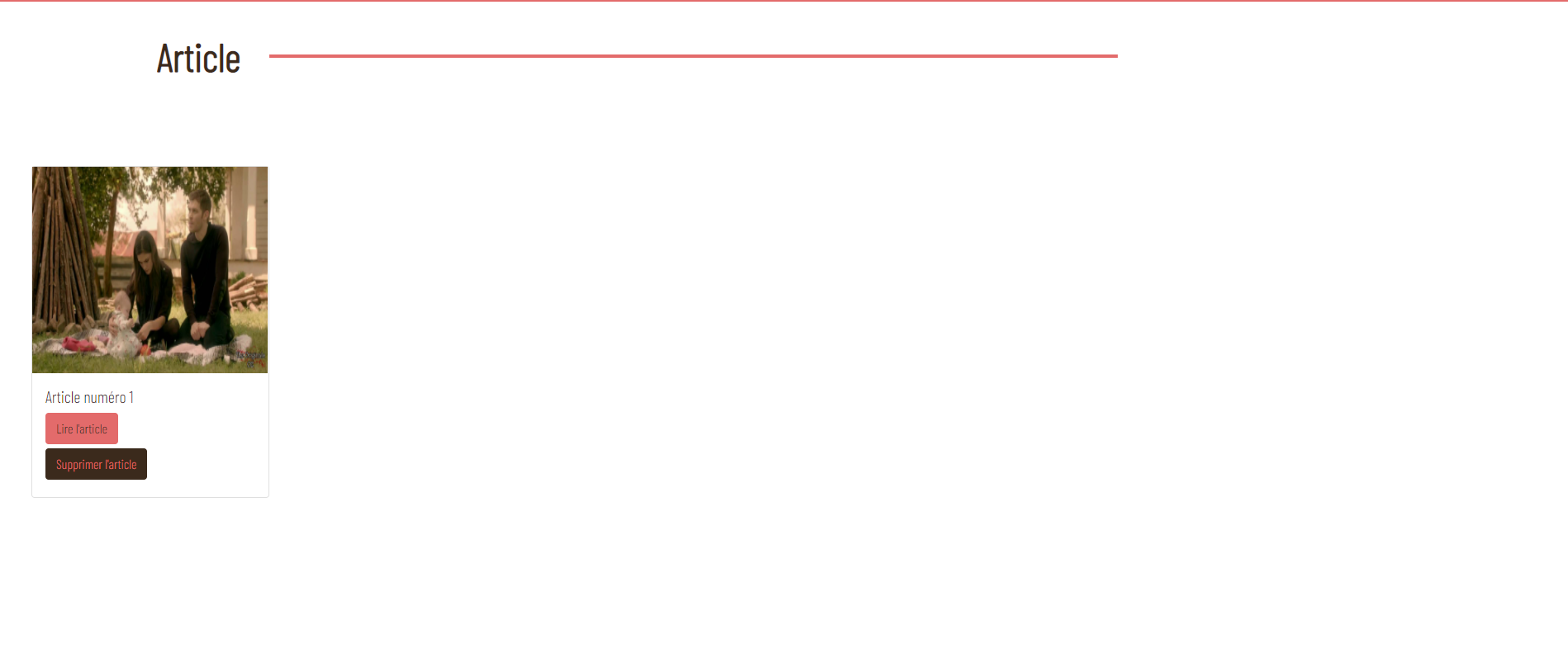
Une image contenant texte

Description générée automatiquement

Pour supprimer l’article j’ai ajouté un onsubmit = « return confirm(‘ …’)

C’est donc sur la vue : display.view.php que la suppression sera affichée. Comme le nom l’indique j’ai retourné une confirmation à l’utilisateur pour lui demander s’il est sûr de vouloir supprimer l’article.

Résultat

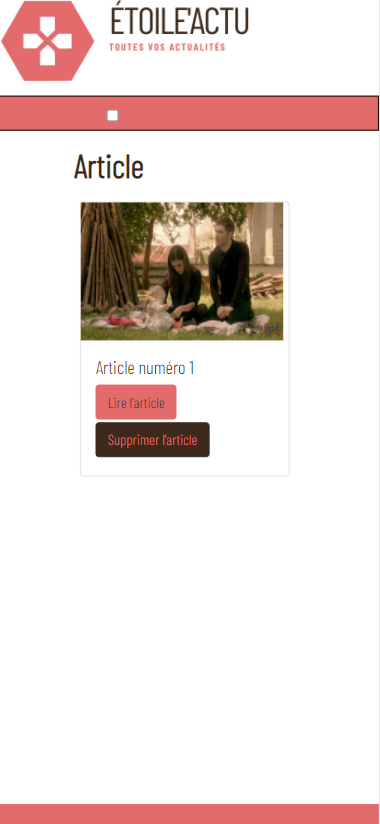


Responsive

Une image contenant texte

Description générée automatiquement

Résultat



1. Back-end
2. Ajout d’article

J’ai opté pour une structure MVC (model, view, controller)

Article Manager addArticleDB

Une image contenant texte

Description générée automatiquement

Comme le titre de la fonction indique, ici, c’est uniquement pour insérer un article dans la base de données. Je mets donc en place une requête préparée afin de sécuriser ma requête. Je précise le nom des colonnes et je leur attribue une « variable ». Je précise donc qu’un nouvel article aura besoin de la base de données et des données sous forme de variable. J’appelle également ma fonction « addArticle($newArticles) » afin de mettre mes données dans un array (tableau).

Global / Article Controller



Une image contenant texte

Description générée automatiquement

Je prépare une fonction « addArticle() » pour mes routes sur mon index afin d’afficher la vue. J’ajoute ensuite une fonction « addArticleValidate » où je précise tout ce dont j’ai besoin dans des variables (emplacement, nom de l’image, date, id). J’appelle ensuite ma fonction du global controller « addImage » dans laquelle j’ai rajouté des contraintes. Une fois que tout ça est vérifié , lorsque le formulaire est envoyé je peux appeler mon manager afin d’appeler ma fonction qui se trouve dans celui-ci en indiquant toutes les informations dont j’ai besoin, les POST proviennent du formulaire, les autres variables ont donc été défini un peu plus haut dans la fonction. Je rajoute un header Location sur ma page d’article pour avoir une redirection.

1. Modifier le profil
2. Supprimer l’article

Article Manager

Une image contenant texte

Description générée automatiquement

Dans mon manager je créer une fonction deleteArticleById($id), pourquoi par l’id ? Tout simplement, sur la vue, lorsqu’on clique sur supprimer, on nous redirige sur une page où j’ai implanté la fonction getIdArticle(), c’est comme ça que chaque article se distingue.

Donc je prépare toujours ma requête, mais cette fois-ci avec un delete où je précise l’id\_article, que je met dans une variable. Si ma requête a fonctionné je stock dans une autre variable mon article par l’id, ensuite je détruis ma variable qui va donc supprimer l’article.

Article Controller

Une image contenant texte

Description générée automatiquement

Dans mon controller, je stock ma fonction du manager vu précédemment et je redirige vers les articles avec un header location.

En revanche, si la requête n’a pas fonctionné donc qu’elle est vide j’envoie « l’article n’existe pas ».

1. **Conclusion**

Ce site est toujours en cours de développement puisque toutes les fonctionnalités ne sont pas prêtes.

Mais, avec ce que j’ai eu la chance d’apprendre, certaines fonctionnalités sont finies et je ne suis pas peu fière du résultat, il me reste néanmoins un long chemin à parcourir et je suis contente d’avoir pu le commencer dans ces circonstances.

J’ai pu remarquer qu’une bonne partie de ce métier était orienté sur la documentation, ce que j’ai vite compris grâce aux méthodes d’apprentissage qui nous poussait à chercher de plus en plus loin puisque les réponses n’étaient pas données.

J’ai rencontré certes des difficultés mais aucun problème qu’on ne pouvait pas résoudre grâce à de la documentation poussée.

Je continuerais, de mon côté, jusqu’à que le site soit fini, pour apprendre un peu plus et développer mes compétences après cette formation.