

# Passion Lecture

P\_Web2 : Page web bibliothèque

Auteurs :

ABIGAËL PÉRISSET

SÉBASTIEN MARC TILLE

VALENTIN PIGNAT

SANTIAGO ESCOBAR TORO

Chef de projet : GRÉGORY CHARMIER

ETML, site de Vennes

Classe FIN27 janvier 2025

Version : 1.0

# 1 TABLE DES MATIÈRES

1	Table des matières .....	2
2	Introduction .....	4
2.1	Fonctionnalités et but.....	4
3	Analyse.....	5
3.1	Éléments communs.....	5
3.2	Accueil.....	6
3.3	Liste des œuvres .....	7
3.4	Modification / Ajout d'un livre .....	9
3.5	Profil.....	10
3.6	Page détail d'œuvre .....	11
3.7	Base de données.....	12
3.7.1	Justification .....	12
3.7.2	Problèmes.....	12
3.7.3	Modèle conceptuel des données (MCD) .....	13
3.7.4	Modèle logique des données (MLD).....	14
3.8	Structure du code.....	15
4	Réalisation .....	16
4.1	Authentification.....	16
4.1.1	Connexion .....	16
4.1.2	Création de compte .....	16
4.1.3	Déconnexion.....	17
4.1.4	Sécurité .....	17
4.2	Reviews .....	17
4.3	Manuel d'utilisation (utilisateur) .....	18
4.3.1	Accueil.....	18
4.3.2	Détail d'œuvre.....	18
4.3.3	Liste et recherche .....	19
4.3.4	Ajout d'œuvres .....	20

4.3.5	Modification d'œuvre .....	23
4.3.6	Suppression d'une œuvre.....	23
4.3.7	Profil utilisateur .....	24
4.3.8	Changement de thème.....	24
4.4	Manuel d'utilisation (admin) .....	24
4.4.1	Utilisation .....	24
4.4.2	Installation.....	25
4.4.3	Connexion à <i>Passion Lecture</i> .....	25
4.4.4	Ajout d'un compte administrateur .....	26
4.5	Ecoconception.....	26
4.6	Inspirations.....	26
5	Tests .....	27
6	Conclusion.....	28
6.1	Générale.....	28
6.2	Personnelle.....	28
6.2.1	Abigaël .....	28
6.2.2	Valentin.....	28
6.2.3	Santiago .....	28
6.2.4	Sébastien.....	28
6.3	Critique .....	29
6.3.1	Pistes d'amélioration .....	29
7	Sources .....	30
8	Glossaire.....	31

## 2 INTRODUCTION

*Passion Lecture* est une application web permettant à ses utilisateurs d'ajouter et noter des livres de différentes catégories. Le présent rapport permet quant à lui de détailler l'aspect technique du projet ainsi que son développement.

D'un point de vue développeur, ce projet vise à mettre en œuvre les connaissances des modules PHP enseignés à l'ETML tout en entraînant les bonnes pratiques de développement (notamment collaboration avec Git, gestion de projet, rédaction de documentation, respect de normes).

### 2.1 Fonctionnalités et but

L'objectif principal de *Passion Lecture* est d'encourager la lecture en permettant à plusieurs utilisateurs de partager leurs trouvailles ou leurs créations et de noter ces dernières. De plus, chaque livre est doté d'un résumé et d'un extrait permettant aux utilisateurs d'obtenir un avant-goût de l'œuvre en question.

## 3 ANALYSE

Ce chapitre décrit et documente les changements apportés aux maquettes durant la phase de développement.

Les maquettes ont été réalisées au tout début du projet. Elles ont servi de base pour le *layout* de *Passion Lecture* mais ne présentent aucune charte graphique.

Les éléments marqués en rouge ne doivent apparaître qu’aux utilisateurs connectés. Seule la partie bleue est accessible au public.

### 3.1 Eléments communs

La navigation et le pied de page ont naturellement été prévus communs à l’ensemble du site. La navigation contient un bouton de retour à l’accueil (le logo), un lien vers la liste des œuvres, un lien vers le formulaire d’ajout ainsi qu’un bouton permettant à l’utilisateur de se connecter.

Si un utilisateur est connecté, le bouton *Connexion* se transforme en bouton de déconnexion et un lien vers le profil est ajouté-

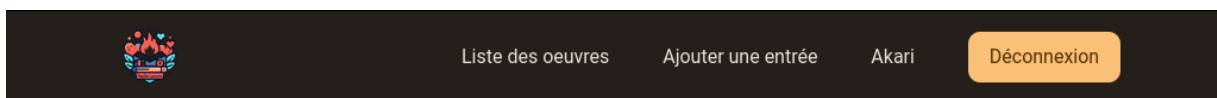


Figure 1. Barre de navigation réalisée. L'utilisateur Akari est connecté.

Dans le *footer*, il était prévu d’indiquer les noms et prénoms des auteurs, la date et la mention ETML. Finalement, quelques éléments ont été ajoutés, notamment un bouton de changement de thème et un lien vers le code source.

Thème	A propos de l'application	Contacteur les développeurs
Changer de thème	Passion Lecture à été développé à l'ETML dans le cadre du module ICT 151. Mis à jour en janvier 2025. Code source	Sébastien Marc Tille Abigaël Périsset Santiago Escobar Toro Valentin Pignat

Figure 2. Footer réalisé.

## 3.2 Accueil

L'accueil contient un petit texte explicatif concernant *Passion Lecture* et une liste des cinq dernières œuvres ajoutées. Dans l'implémentation, le format a été respecté et les changements n'ont été que mineurs (alignements).

La liste se présente sous forme de cartes qui contiennent l'image de couverture du livre ainsi que les informations de bases du livre (titre, auteur, genre, note moyenne, membre ayant ajouté l'ouvrage).

Bien que toujours présent, le bouton *Détail* n'est en réalité pas très utile car l'ensemble de la carte est cliquable.

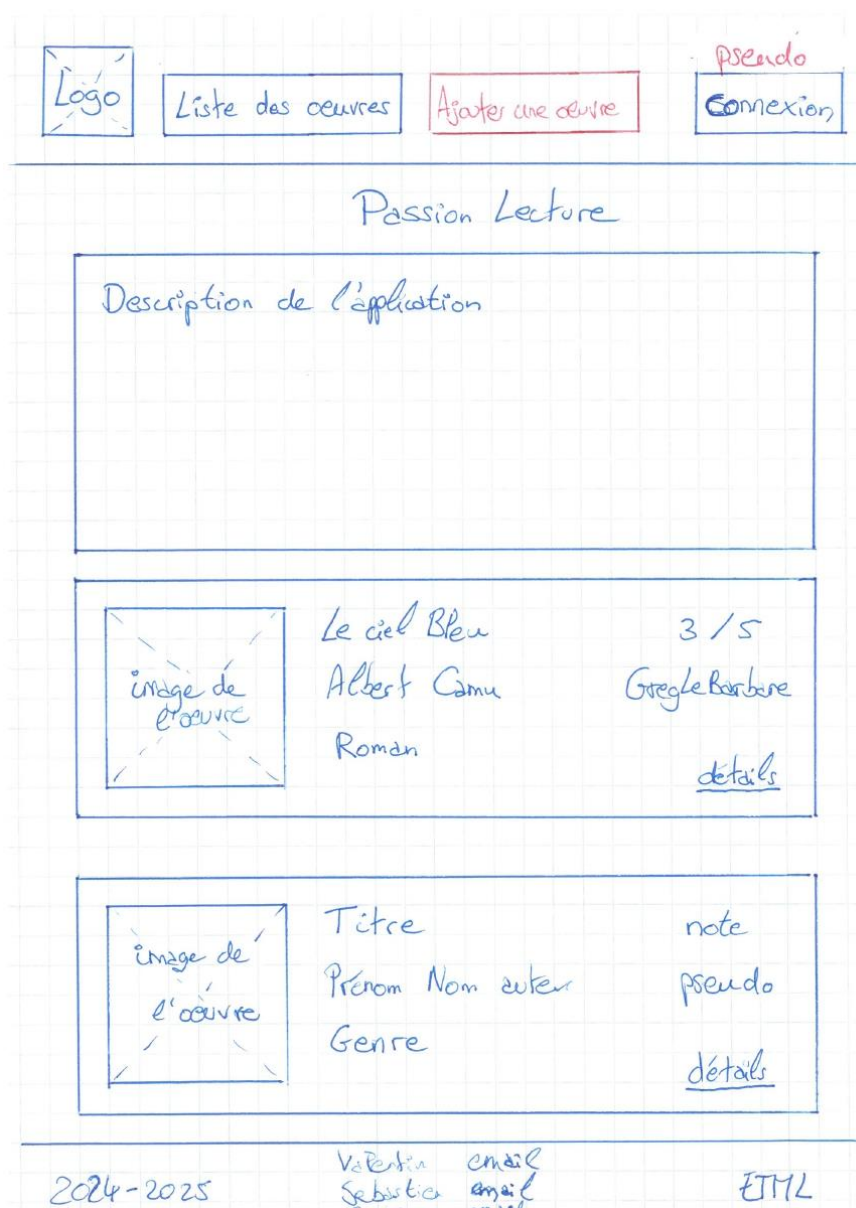


Figure 3. Maquette de l'accueil.

### 3.3 Liste des œuvres

La liste des œuvres comporte une barre de recherche ainsi qu'une liste déroulante permettant de filtrer par genre. Contrairement à la page d'accueil, ici tous les livres sont présents.

Logo

Liste des œuvres

Ajouter une œuvre

pseudo  
Connexion

---

Liste des œuvres

genre

liste déroulante

recherche

image de l'œuvre	Le ciel Bleu Albert Camus Roman	3/5 Gegele Barbare <u>détails</u>
image de l'œuvre	Titre Prénom Nom auteur Genre	note pseudo <u>détails</u>

idem l'affichage pour toutes les œuvres

footer identique

Figure 4. Maquette de la liste des œuvres.

La réalisation a été légèrement différente. Au lieu d'afficher toutes les œuvres en une fois, des pages de 10 éléments ont été mises en place. Aussi, la carte a été légèrement modifiée pour être plus compact et rendre la navigation plus simple.

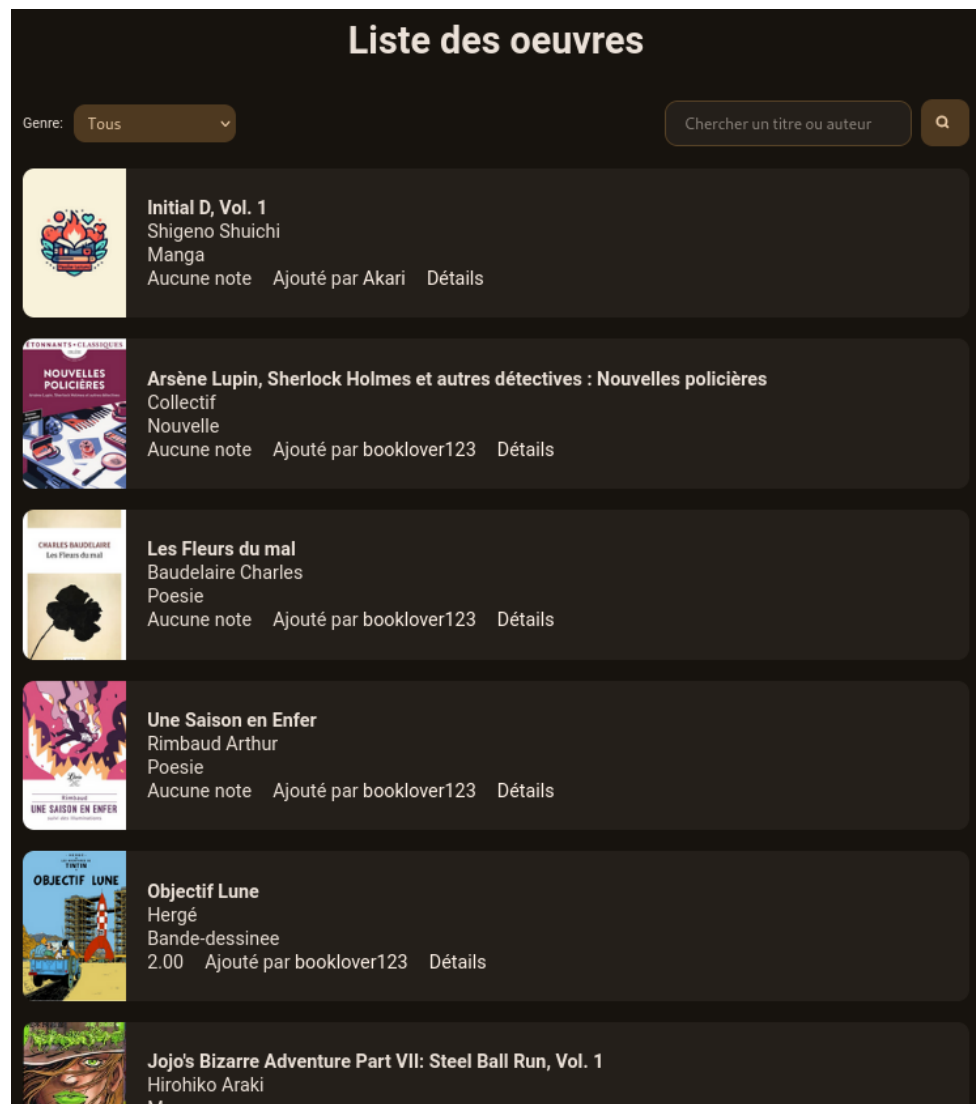


Figure 5. Liste réalisée.



### 3.4 Modification / Ajout d'un livre

Le formulaire d'ajout est réutilisé pour la modification. Tous les champs sont nécessaires.

Logo Liste des oeuvres Ajouter une oeuvre pseudo déconnexion

Modification  
Ajout d'un livre

Nom de l'auteur :

Prénom de l'auteur :

Titre du livre :

Editeur :

Année d'édition :

Nombre de pages :

Genre :

Image de couverture :

Lien vers un extrait :

Résumé :

Footer identique

Figure 6. Maquette du formulaire d'ajout et de modification.

Ajout d'un livre

Nom de l'auteur

Prénom de l'auteur

Titre du livre

Editeur

mm / dd / yyyy    
Année d'édition

Nombre de page

Genre  
bande-dessinée

Image de couverture  
 No file selected.

Lien vers un extrait

Résumé

Figure 7. Formulaire d'ajout / de modification réalisé.

### 3.5 Profil

Cette page sert à afficher les détails qui concernent l'utilisateur, notamment :

- La date de création du compte
- Le nombre et la liste des livres ajoutés
- Le nombre et la liste des livres notés

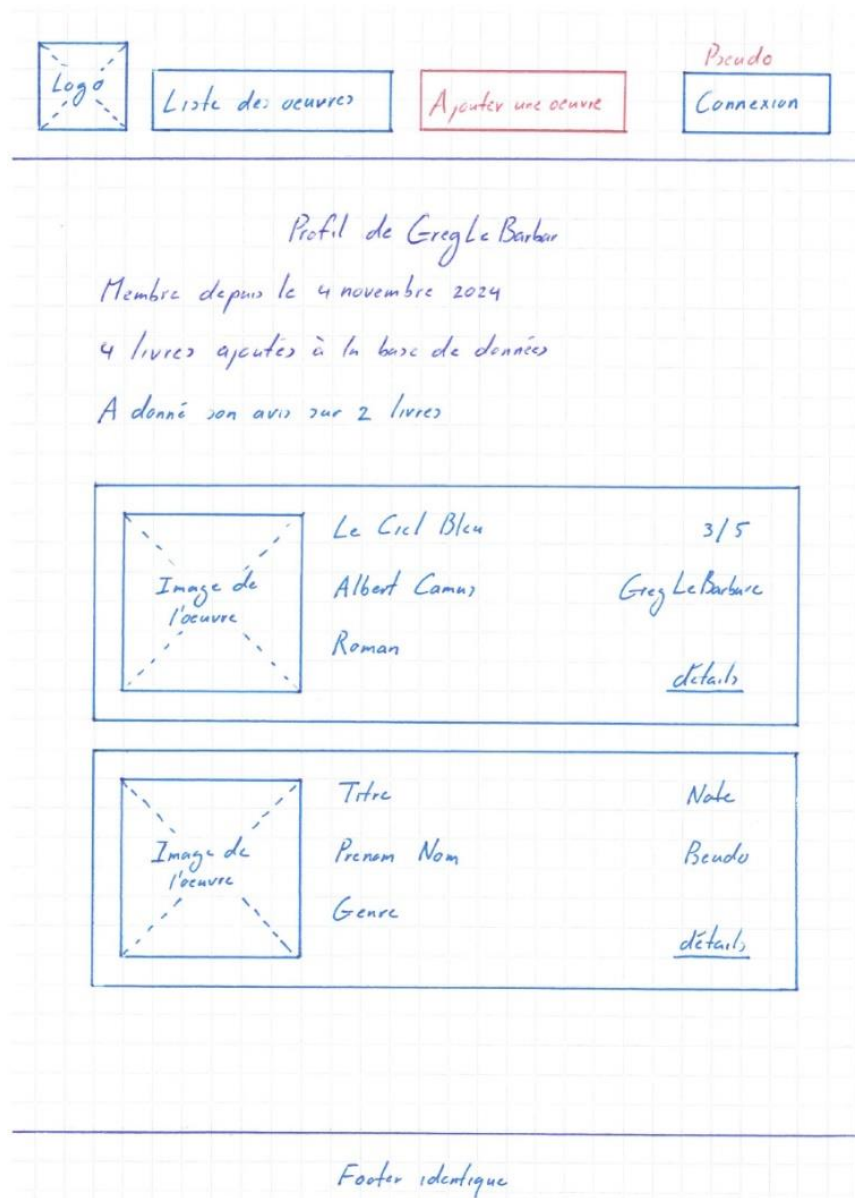


Figure 8. Maquette de la page de profil.

La seule différence notable a été l'utilisation de la version compacte des cartes de livres et l'ajout d'un *placeholder* pour la photo de profil, en prévision d'un éventuel ajout de la gestion de celles-ci.

### 3.6 Page détail d'œuvre

Cette page, accessible uniquement pour les utilisateurs connectés, sert à afficher les détails concernant un livre.

Il était prévu de faire un système de notation avec des étoiles, mais au vu de la difficulté quant à une réalisation « à la main », il a été décidé d'utiliser une simple liste déroulante.

Logo

Liste des œuvres

Ajouter une œuvre

pseudo

Connexion

---

Détail de l'œuvre

image de l'œuvre

Pseudonyme Nom auteur

Titre

Editeur année édition

★ ★ ★ ★ ★

Valider

moyenne des notes

Genre

Nombre de pages

Utilisateur

Extrait

Résumé

---

footer identique

## 3.7 Base de données

La base de données est dotée des [entités](#) suivantes :

- `t_author` pour représenter les auteurs
- `t_publisher` pour représenter les éditeurs
- `t_book` pour représenter les livres
- `t_category` pour représenter les genres
- `t_user` pour représenter les utilisateurs

Ensuite, il a été décidé<sup>1</sup> que :

- Un auteur peut avoir de 0 à n livres et un livre ne peut avoir qu'un seul auteur.
- Un éditeur peut publier de 1 à n livres et un livre ne peut avoir qu'un seul éditeur.
- Un utilisateur peut ajouter de 0 à n livres et un livre ne peut être ajouté que par un seul utilisateur.
- Un livre ne peut avoir qu'une seule catégorie et chaque catégorie peut avoir de 0 à n livres (les catégories sont prédéfinies; par défaut, il n'y a aucun livre).
- Un utilisateur peut évaluer de 0 à n livres et un livre peut être évalué par de 0 à n utilisateurs.

La base est compatible avec MySQL et MariaDB.

### 3.7.1 Justification

Ce modèle permet une bonne flexibilité et évolutivité du programme. En ayant séparé les auteurs des livres et des éditeurs, il est possible de créer des [filtres](#) de recherche plus fine et d'éviter de potentielles migrations si, un jour, il deviendrait nécessaire d'ajouter des pages de détails pour les auteurs ou les éditeurs (à l'image de ce que fait *Anilist*) ou d'augmenter les *reviews* de commentaires.

Il a aussi permis de faciliter les requêtes complexes [comment], telles que la recherche de livre par catégories ou le calcul de la moyenne.

Les tailles des champs ont été choisies en se référant à des projets existants ou en s'informant sur les bonnes pratiques actuelles (notamment en ce qui concerne les 2048 caractères de l'URL<sup>2</sup>). Le champ `password` a été limité à 60 caractères car l'output de la méthode `password_hash` retourne toujours une chaîne de 60 caractères.

### 3.7.2 Problèmes

Lors du développement, il a été remarqué que, sous *Windows*, les accents n'apparaissent pas toujours, même au travers du navigateur. Dans un environnement Linux, ce problème disparaît. Malheureusement, par manque de temps, il n'a pas été possible d'en trouver la source.

---

<sup>1</sup> Mise au commun avec le professeur et la classe.

<sup>2</sup> <https://stackoverflow.com/questions/417142/what-is-the-maximum-length-of-a-url-in-different-browsers>

### 3.7.3 Modèle conceptuel des données (MCD)

Voici le modèle [MCD](#). Ce dernier a été réalisé en classe par l'ensemble des élèves afin d'avoir une base commune dans le projet. Ainsi les types des données et leur taille ont été définies en amont de la création du projet. Notre base de données reprend les mêmes types et tailles de la figure ci-dessous. Cependant, nous avons décidé de modifier la taille de l'url, le champ « excerpt » dans la table « BOOK ». En effet, pour appliquer une bonne pratique de la taille maximum de l'URL<sup>1</sup>, nous avons fixé la limite à 2048 caractères.

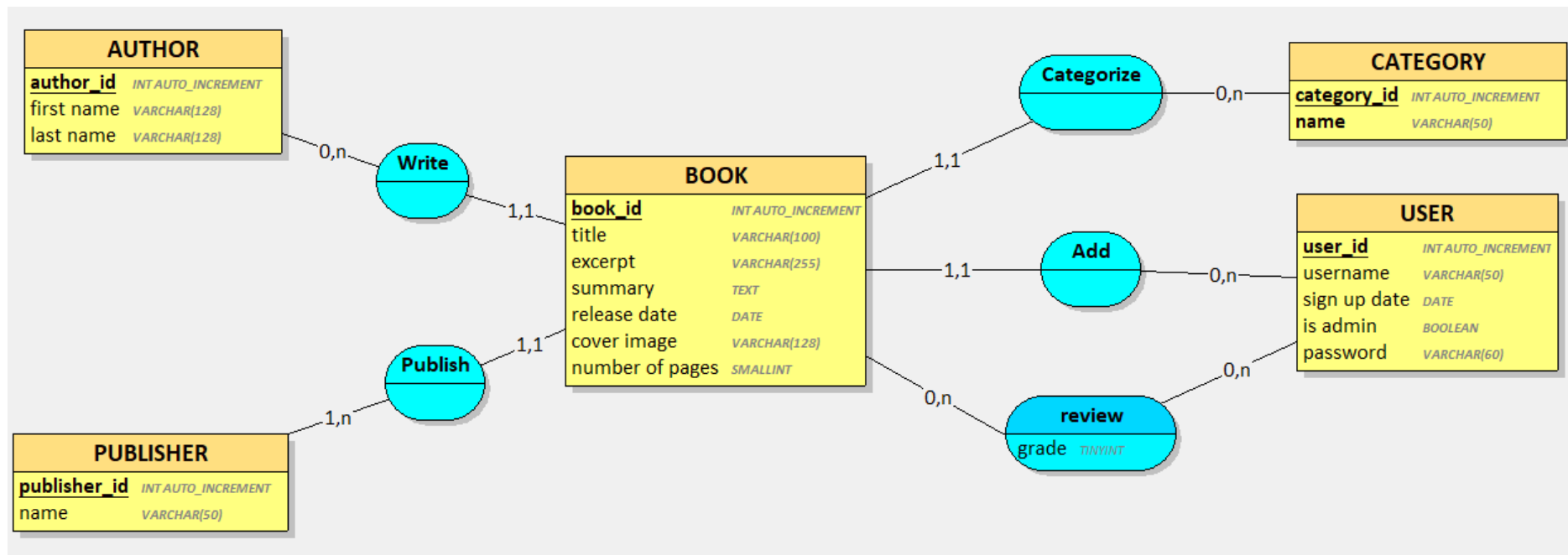


Figure 9. MCD généré par Looping.

<sup>1</sup> <https://stackoverflow.com/questions/417142/what-is-the-maximum-length-of-a-url-in-different-browsers>

### 3.7.4 Modèle logique des données (MLD)

Voici le modèle [MLD](#) de la base de données dans lequel vous pouvez voir l'entité livre et une relation avec les clés principales et étrangères.

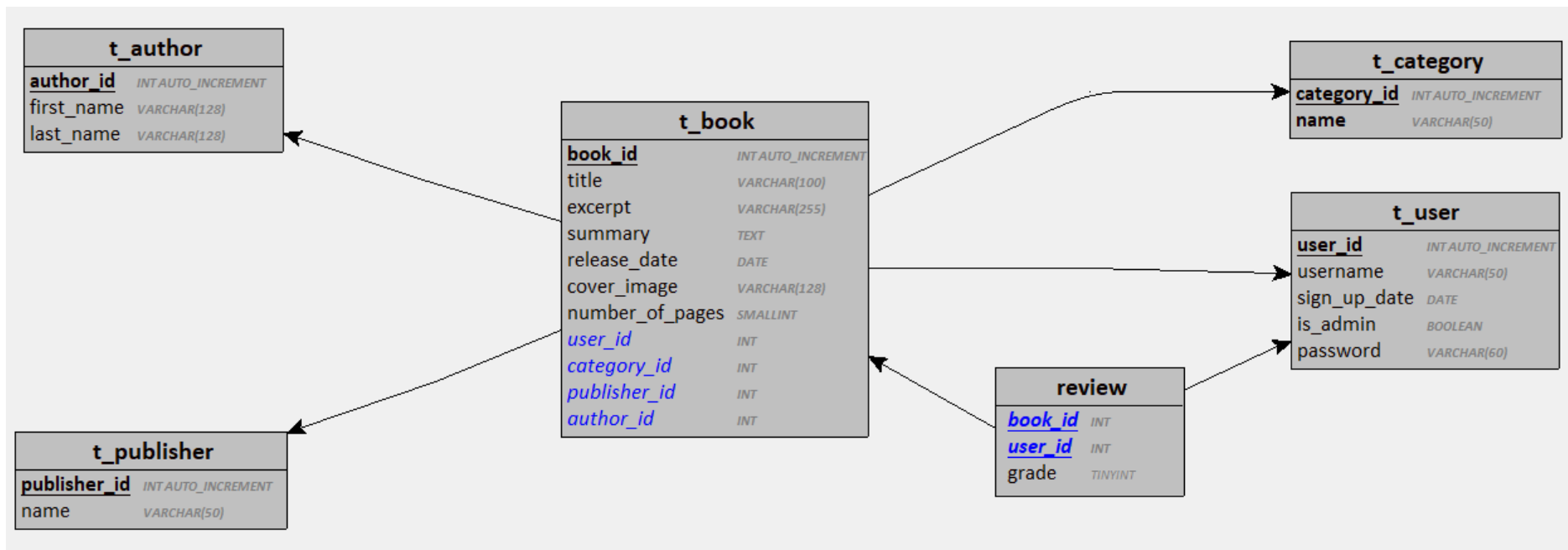
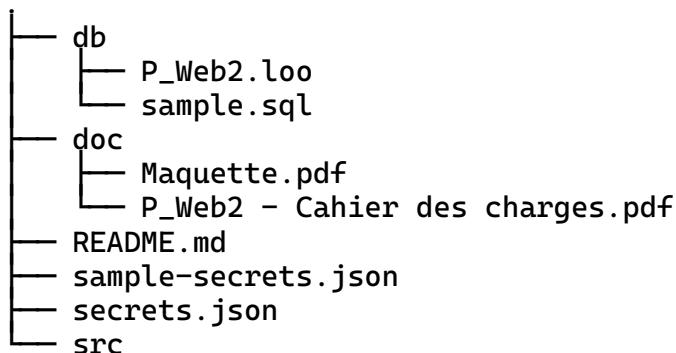


Figure 10. MLD généré par Looping.

## 3.8 Structure du code

*Passion Lecture* a été conçu en PHP en suivant le *design pattern* MVC. Le haut de l'arborescence se présente ainsi :



Le dossier **db** contient le diagramme *Looping* de la base de données ainsi qu'un *dump* contenant quelques entrées et comptes utilisateurs<sup>1</sup> (**sample.sql**).

**doc** contient un document qui reprend toutes les maquettes ainsi que le cahier des charges.

Le dossier **src** contient l'ensemble du code source de *Passion Lecture* (aussi accessible sur *GitHub*<sup>2</sup>). À noter que les styles ont été rédigés en SCSS et qu'ils doivent être compilés. Une configuration a été préparée et se trouve dans le dossier caché **.vscode**. Cette décision a été motivée par un CSS devenu trop long et compliqué à maintenir : il a été découpé en morceau en suivant le [pattern 7-1](https://sass-guidelin.es/#the-7-1-pattern)<sup>3</sup>.

Les fichiers PHP sont répartis dans les dossiers **controllers**, **helpers**, **models** et **views**, selon leur utilité. Pour éviter la duplication de code inutile, des *partials* ont été créés pour les vues (pour le *footer*, la barre de navigation et l'élément **head** des pages HTML / PHP).

Le fichier **sample-secrets.json** contient un exemple de configuration des identifiants de connexion à la base de données (à renommer en **secrets.json**).

---

<sup>1</sup> Les mots de passe sont toujours **aaa**.

<sup>2</sup> <https://github.com/wolfiyy/p-web2>

<sup>3</sup> <https://sass-guidelin.es/#the-7-1-pattern>

## 4 RÉALISATION

Ce chapitre détaille les aspects techniques des fonctionnalités d'authentification, de notation d'œuvres ainsi qu'un manuel destiné aux administrateurs.

### 4.1 Authentification

Le système d'authentification s'inspire de nombreux sites existants. En cliquant sur le bouton *Connexion* dans la barre de navigation, un formulaire à deux champs (*Nom d'utilisateur* et *Mot de passe*) apparaît.

#### 4.1.1 Connexion

Depuis ce formulaire, un utilisateur existant peut s'identifier en entrant ses informations de connexion. Si les informations de connexion sont erronées, l'utilisateur est prévenu et il lui est demandé de les entrer à nouveau.

La validation se fait en comparant l'entrée utilisateur avec le [hash](#) du mot de passe enregistré dans la base de données en se servant de la méthode PHP `password_verify`.

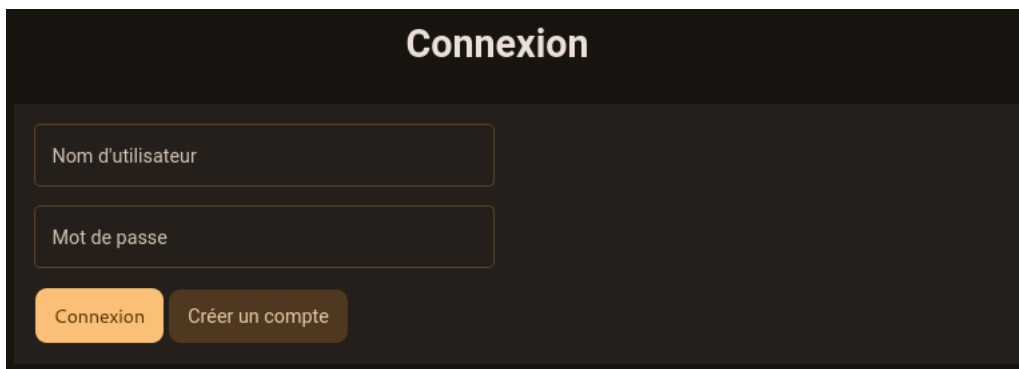


Figure 11. Formulaire de connexion.

Une fois les données vérifiées et validées, trois variables de session sont créées :

1. **user\_id** – Permet de stocker l'identifiant unique de l'utilisateur.
2. **username** – Permet de stocker le nom d'utilisateur.
3. **is\_admin** – Permet de stocker le niveau de permissions du compte (0 pour les utilisateurs standard et 1 pour les administrateurs).

Finalement, l'utilisateur est redirigé sur son profil.

#### 4.1.2 Création de compte

Si l'utilisateur désire créer un nouveau compte, il peut le faire en cliquant sur le bouton *Créer un compte* qui l'emmène sur une autre page contenant trois champs : *Nom d'utilisateur*, *Mot de passe* et *Mot de passe (confirmation)*.



Si un nom d'utilisateur existant est utilisé, ou si les mots de passes ne sont pas les mêmes, un message d'erreur apparaît et il est demandé à l'utilisateur de corriger la ou les erreurs.



**Création d'un compte**

Veuillez renseigner les champs suivants.

Nom d'utilisateur

Mot de passe

Mot de passe (confirmation)

Créer le compte

Figure 12. Formulaire de création de compte.

A noter que le nom d'utilisateur est insensible à la case (i.e. booklover123 peut se connecter en tant que BOOKLOVER123). Il n'est donc pas possible d'avoir deux comptes bookLOVER123 et BOOKlover123.

### 4.1.3 Déconnexion

Lorsqu'un utilisateur est connecté, le bouton de la barre de navigation se transforme en bouton de déconnexion. Quand celui-ci est cliqué, la session est détruite et l'utilisateur est redirigé vers la page de connexion<sup>1</sup>.

### 4.1.4 Sécurité

Les œuvres ne peuvent être modifiées que par les personnes les ayant ajoutée ou un administrateur. Une vérification est effectuée dans le code pour masquer les boutons si ces deux conditions n'est pas respectée.

De plus, et selon le cahier des charges, l'affichage des détails d'œuvres ne peut se faire que pour les utilisateurs connectés.

## 4.2 Reviews

Les utilisateurs connectés ont la possibilité de laisser une note sur chaque livre. La moyenne d'un livre est calculée en prenant en compte tous les utilisateurs ayant noté celui-ci. De plus, depuis la page de profil d'un utilisateur, toutes les notes attribuées sont visibles. Lors de la suppression d'un livre, les notes attribuées sont retirées de la base de données.

---

<sup>1</sup> Idéalement, l'utilisateur devrait être redirigé vers la page d'accueil. Par manque de temps, il n'a pas été possible de corriger ce problème.



Moyenne  
2.00

Nombre d'appréciations  
1

Genre  
Bande-dessinée

## Détails de l'oeuvre

### Objectif Lune

Auteur: Hergé

Date de sortie: 1 January 1950

Editeur: Casterman

Votre note:  Valider

#### Résumé

Alors qu'ils reviennent à Moulinsart après leur périple en Afrique du Nord (voir "Au pays de l'or noir"), Tintin et Haddock apprennent que Tournesol a quitté le château voilà trois semaines... Quelques minutes plus tard, Nestor leur amène un télégramme du professeur, qui leur demande de le rejoindre en Syldavie (le pays de l'eau minérale, un comble pour le capitaine !!). Une fois arrivée à l'aéroport, ils sont réceptionnés par un ami de Tournesol qui les conduit, à travers un impressionnant dispositif de contrôle, jusqu'à une base secrète. Une fois arrivé, Tournesol leur explique qu'il dirige la section aéronautique du Centre de Recherches Atomiques de Sbrodj, et qu'à ce propos il projette d'envoyer une fusée sur la lune, dont Tintin, Haddock et lui seraient les passagers ! Un soir, l'alarme de la base se déclenche : des parachutistes ont été lâchés au dessus du centre. Des espions ? Tintin décide d'enquêter...

L'utilisateur peut évaluer un livre depuis la page de détail en sélectionnant une note et en appuyant sur valider. Sur cette page et dans la liste des œuvres, la moyennes des évaluations est affichée.

## 4.3 Manuel d'utilisation (utilisateur)

Le fonctionnement de sites est très similaire à celui de [plateformes existantes](#) comme *Goodreads*, *RateYourMusic*, ou *Anilist*.

### 4.3.1 Accueil

La page d'accueil affiche un texte de présentation ainsi qu'une liste contenant les cinq derniers ouvrages à avoir été ajoutés. Les éléments de celles-ci peuvent être cliqués pour accéder à des informations détaillées.

### 4.3.2 Détail d'œuvre

La page de détail affiche un résumé et quelques informations sur le livre (notamment le genre et le nombre de pages). Un lien vers un extrait peut y être trouvé et l'image de couverture peut être ouverte en cliquant dessus.

Pour ajouter une note, il suffit de sélectionner la note désirée sur le menu déroulant et de cliquer le bouton *Valider*.

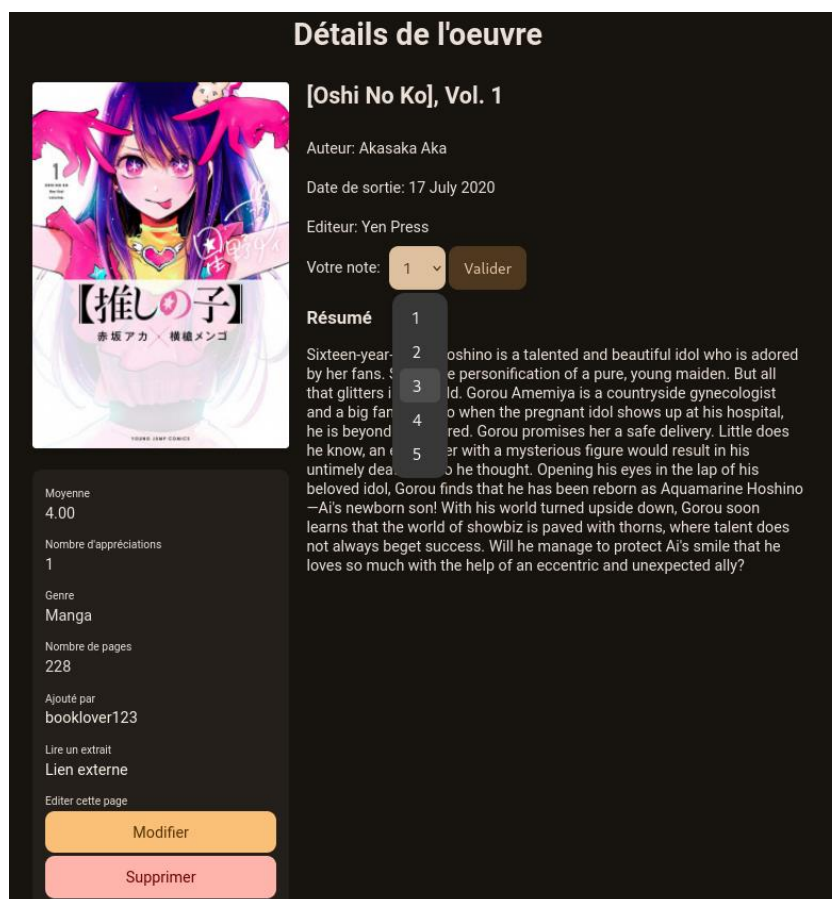


Figure 13. Page de détail d'un livre. L'utilisateur est en train de mettre une note.

Si l'utilisateur ayant ajouté l'œuvre est connecté, des boutons *Modifier* et *Supprimer* s'affichent.

#### 4.3.3 Liste et recherche

La page *Liste des œuvres* affiche l'entièreté des œuvres enregistrées dans la base de données de *Passion Lecture* par ordre d'ajout (le plus récent en premier). Un menu déroulant permet de filtrer par genre et un champ permet d'effectuer une recherche par auteur ou par titre.

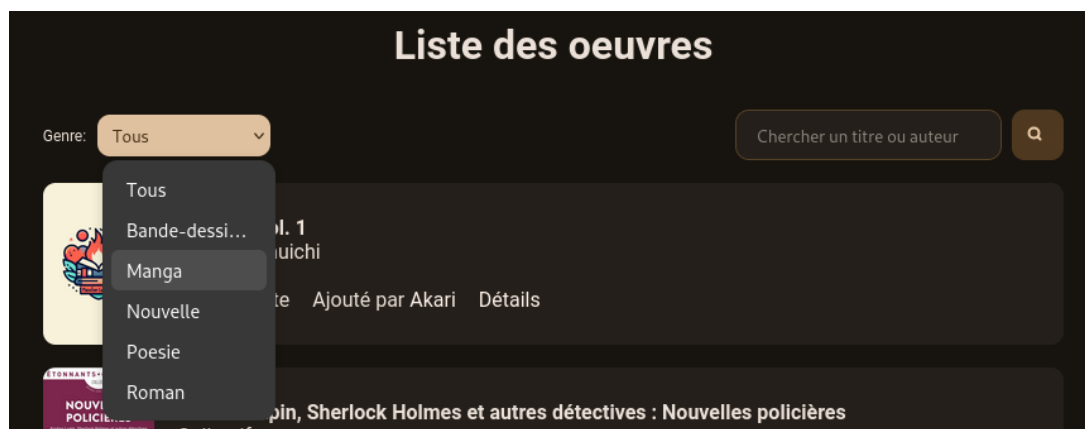


Figure 14. Filtres de recherche.

#### 4.3.4 Ajout d'œuvres

Le lien *Ajouter une œuvre*, disponible depuis la barre de navigation permet d'ajouter une nouvelle œuvre à *Passion Lecture*.

En le cliquant, un grand formulaire apparaît avec des champs tous obligatoire :

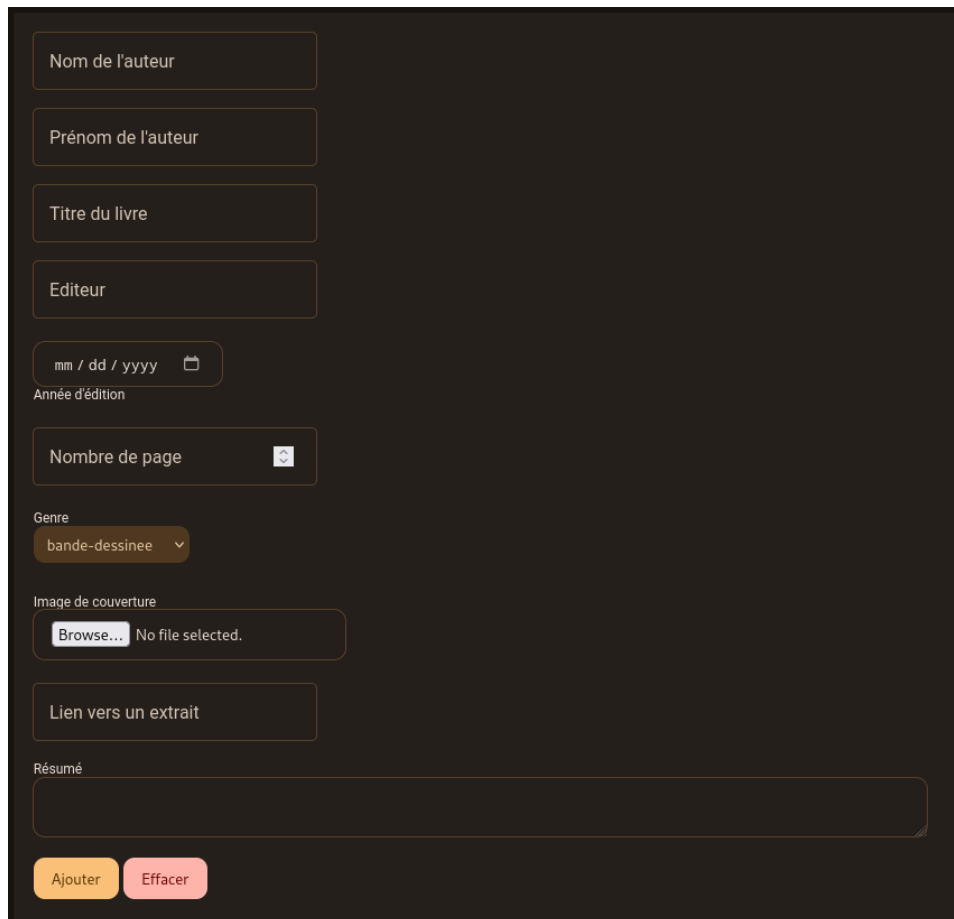
- Nom de l'auteur : au maximum 128 caractères.
- Prénom de l'auteur : au maximum 128 caractères.
- Titre du livre : au maximum 100 caractères.
- Editeur : au maximum 50 caractères.
- Date de parution : peut être sélectionné à l'aide du petit calendrier. En cas d'entrée manuelle, faire attention au format de date qui dépend de l'OS !
- Nombre de pages : doit être un nombre entier positif.
- Genre : à sélectionner dans la liste prédéfinie.
- Image de couverture : une image à utiliser pour représenter le livre. Elle doit être au format **jpeg** ou **png**.
- Lien vers un extrait : doit être une URL valide de maximum 2048 caractères.
- Résumé : un court résumé de maximum 2000 caractères.

Une fois le bouton *Valider* est cliqué, si une ou plusieurs valeurs invalides sont entrées, un ou plusieurs messages d'erreur(s) s'affichent en dessus des champs concernés. Les valeurs entrées dans le formulaire sont réaffichées après l'échec de la validation à l'exception de l'input pour l'image car le chemin du disque ne peut pas être récupéré et utilisé comme défaut pour le champ<sup>1</sup>.

Enfin, le bouton *Effacer* permet de vider le formulaire.

---

<sup>1</sup> <https://stackoverflow.com/questions/1696877/how-to-set-a-value-to-a-file-input-in-html-to-a-client-side-disk-file-system-pat>




Nom de l'auteur


Prénom de l'auteur

Titre du livre

Editeur

mm / dd / yyyy 

Année d'édition

Nombre de page 

Genre

bande-dessinee ▾

Image de couverture

No file selected.

Lien vers un extrait

Résumé

Figure 15. Formulaire d'ajout d'œuvre.

#### 4.3.4.1 Contrôle des données

En cliquant sur le bouton ajouter, les données saisies par l'utilisateur vont être contrôlée et on va appliquer des filtres de protection contre les injections SQL ou les scripts malveillants.

Premièrement chaque donnée saisie par l'utilisateur va passer par les « [filter sanitize](#) » propre à PHP. Ainsi tous les caractères spéciaux sont représentés par des [entités HTML](#). Pour ce qui est de l'URL, il s'agit aussi d'un filtre qui va nettoyer la chaîne, puis on applique un second filtre qui contrôle si l'URL est valide selon » [RFC 2396](#)<sup>1</sup>. Pour le champ du nombre de page, il s'agit d'un filtre qui retire toute valeurs non numériques. Ensuite on applique un second contrôle qui n'accepte que les nombres positifs plus grand que 0.

En deuxième lieu, on contrôle que chaque champ a été remplis, sinon une erreur apparait sous chaque champ vide. De plus, on va appliquer un contrôle sur la longueur des champs texte et varchar. Ce contrôle permet que aucune donnée ne dépasse la taille limite de notre base de données. Si un champ contient une taille qui ne respecte pas la taille maximal ou minimum autorisé, une erreur s'affiche en dessous du champ.

Lors de ce contrôle de données nous avons fait le choix de laisser à l'utilisateur la totale liberté des caractères saisis dans les champs. En effet nous avons eu des problèmes avec

<sup>1</sup> <https://www.php.net/manual/en/filter.constants.php>

l'application de [regex](#) comme contrôleurs. Ces derniers laissent passer l'injection SQL et les scripts avant de passer dans le contrôle de caractères. Nous avons essayé de mettre des filtres puis les regex, mais comme les caractères spéciaux sont traduits en entité HTML, l'utilisation des regex devient trop compliquée pour différencier un caractère autorisé d'un caractère non autorisé. C'est pourquoi, l'utilisateur peut entrer n'importe quel caractère dans les champs du formulaire. Ce choix diminue la sécurité concernant les valeurs saisies dans le sens que des fausses informations peuvent être saisies, par exemple, « 1234 » pour le nom de l'auteur, mais protège le formulaire des attaques SQL et des scripts malveillants.

En dernier lieu vient la validation de l'image entrée par l'utilisateur. Il faut préciser que nous contrôlons que l'image ne dépasse pas le 2 Mo en taille et que nous contrôlons l'extension de fichiers pour accepter uniquement les «.png» et «.jpeg». De plus nous avons convenu que lorsqu'une erreur survient dans la validation du formulaire et qu'il ne s'agit pas de l'image, l'image de couverture n'est pas gardée en mémoire. Ce qui veut dire que l'on demande l'ajout de l'image dans le formulaire à chaque fois que l'utilisateur ne respecte pas une règle de validation. Ce choix a été pris car si l'image ne présente pas d'erreur donc elle est validée et ajoutée à la base de données et que l'utilisateur décide d'effacer le formulaire, nous aurions dû prévoir des fonctionnalités supplémentaires pour aller supprimer le chemin d'accès dans la base de données. Nous avons décidé de ne pas ajouter ces fonctionnalités par manque de temps et de nous concentrer sur d'autres fonctionnalités de notre site web.

Figure 16. Messages d'erreurs

Cette action est uniquement disponible sur les œuvres ajoutées par le compte connecté.

Pour la vérification des champs, le code est repris des ajouts de livre. Certaines

Cette action est uniquement disponible sur les œuvres ajoutées par le compte connecté.

### 4.3.7 Profil utilisateur

La page de profil permet d’afficher la date d’inscription ainsi qu’une liste des ouvrages ajoutés et notés. Si aucun livre n’a été noté ou ajouté, les listes sont masquées.

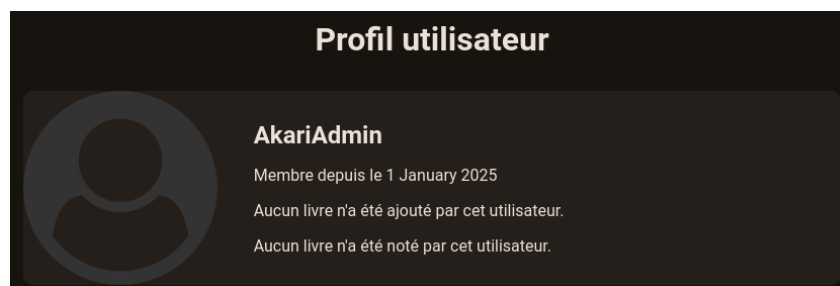


Figure 17 17. Profil vide.

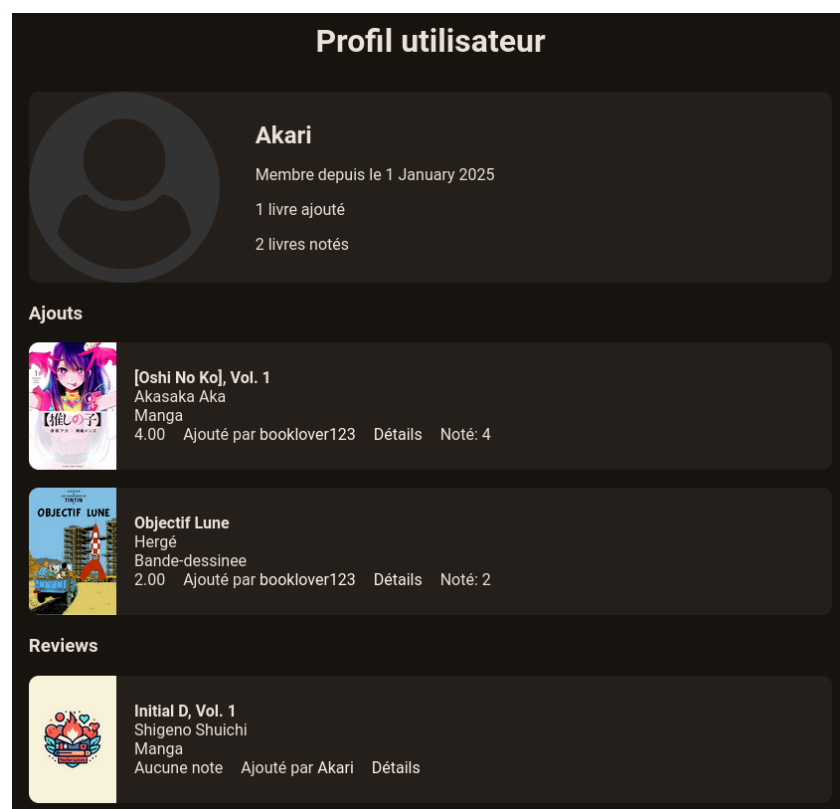


Figure 18 18. Profil ayant ajouté deux livres et noté un livre.

### 4.3.8 Changement de thème

L’application est disponible en thèmes clair et sombre. Le changement peut se faire depuis le pied de page en cliquant sur le lien *Changer de thème*.

## 4.4 Manuel d’utilisation (admin)

### 4.4.1 Utilisation

Pour l’utilisation « basique » de l’application, se référer au chapitre précédent (*Manuel d’utilisation (utilisateurs)*).



Sur l'application, la seule différence visible pour les administrateurs est l'ajout de boutons *Modifier* et *Supprimer* sur l'ensemble des pages de détail d'œuvres. En effet, les administrateurs ont le contrôle total sur toutes les entrées.

#### 4.4.2 Installation

PHP doit être installé<sup>1</sup>. *Passion Lecture* peut être cloné depuis son dépôt Git public :

```
git@github.com:wolfiiy/p-web2.git
```

##### 4.4.2.1 Mise en place de la base de données (Linux)

Malheureusement, il n'y a pas de script de mise en place de la base de données. Celle-ci doit être créée manuellement depuis une console MySQL ou MariaDB.

```
cd p-web2/db
```

```
sudo mariadb
```

```
create database db_passion_lecture;
```

Ensuite, toujours depuis la console du SGBD<sup>2</sup>, il est recommandé de créer un nouvel utilisateur pour interagir avec *Passion Lecture*.

```
create user 'utilisateur'@localhost identified by 'mot-de-passe';
```

```
grant all on db_passion_lecture.* to 'utilisateur'@localhost with grant option;
```

Ensuite, la base de données d'exemple peut être importée dans MySQL ou MariaDB en utilisant la console appropriée.

```
use db_passion_lecture;
```

```
source sample.sql
```

```
create database db_passion_lecture;
```

##### 4.4.2.2 Configuration des secrets

Enfin, les identifiants précédemment créés au travers de la console du SGBD doivent être reportés dans un fichier **secrets.json**. Un exemple est disponible à la racine du projet sous le nom **sample-secrets.json**.

Le port et le nom d'hôte peuvent être ajustés si besoin.

#### 4.4.3 Connexion à *Passion Lecture*

Quelques utilisateurs ont été préconfigurés :

---

<sup>1</sup> PHP 8.2.26, 8.3.10 et 8.3.15 testés.

<sup>2</sup> Système de gestion de base de données.

user_id	username	sign_up_date	is_admin	password
1	booklover123	2022-01-15	0	\$2y\$10\$QdMUZKCYXFi08vitCY8F80ez59mM.xe7pPMB6v8r1T0ZuIxwmEk.S
2	admin_guy	2021-12-05	1	\$2y\$10\$QdMUZKCYXFi08vitCY8F80ez59mM.xe7pPMB6v8r1T0ZuIxwmEk.S
3	reading_queen	2023-03-22	0	\$2y\$10\$QdMUZKCYXFi08vitCY8F80ez59mM.xe7pPMB6v8r1T0ZuIxwmEk.S
5	Akari	2025-01-01	0	\$2y\$10\$QdMUZKCYXFi08vitCY8F80ez59mM.xe7pPMB6v8r1T0ZuIxwmEk.S
6	AkariAdmin	2025-01-01	1	\$2y\$10\$UwWJv.kZ2UCBd8Y0H7huSu2nWZIQTIrbYDNao0AyxFqVw8i6noVi6

Les mots de passe sont toujours « aaa ».

#### 4.4.4 Ajout d'un compte administrateur

Si un compte administrateur doit être créé, il est recommandé de créer un compte au travers du formulaire dans l'application puis de manuellement changer le niveau de permission de l'utilisateur en question depuis la console MySQL / MariaDB.

```
update t_user set is_admin = 1 where user_id = x;
```

Où x doit être remplacé par l'ID de l'utilisateur à promouvoir administrateur.

## 4.5 Ecoconception

*Passion Lecture* respecte quelques bonnes pratiques quant à la [conception écologique](#) d'une application web, notamment le fait de préférer la pagination au défilement infini<sup>1</sup> (cf. liste des livres).

## 4.6 Inspirations

La palette de couleur a été générée en utilisant l'outil *Figma* de *Google* pour le *Material Design 3*.

Le code permettant d'afficher et de masquer la barre de navigation provient d'un projet précédemment réalisé par l'un des développeurs<sup>2</sup>. Il en va de même pour la gestion du thème sombre<sup>3</sup>.

Les *inputs* ont été très fortement inspirés par le travail d'une personne tierce, LeonGeldsch sur *GitHub*<sup>4</sup>.

Les données de la base d'exemple proviennent de *MyAnimeList* et de *GoodReads*.

<sup>1</sup> Numéro 13, [https://collectif.greenit.fr/ecoconception-web/115-bonnes-pratiques-eco-conception\\_web.html](https://collectif.greenit.fr/ecoconception-web/115-bonnes-pratiques-eco-conception_web.html)

<sup>2</sup> <https://gitlab.com/wolfiy/etml-c101-projet>

<sup>3</sup> <https://gitlab.com/wolfiy/wlfys-minimal-startpage>

<sup>4</sup> [https://github.com/LeonGeldsch/css\\_material\\_input](https://github.com/LeonGeldsch/css_material_input)

## 5 TESTS

Seule une classe de tests unitaires a été créée, **FormatHelperTests**. Cette classe permet de tester la class statique **FormatHelper** qui s'occupe de convertir les dates du format [ISO 8601](#) (type 2025-01-07) en une chaine de caractères (type 7 janvier 2025).

Les tests ont été réalisés à l'aide de [PHPUnit](#). Celui doit être installé sur le système car *Passion Lecture* n'a aucune dépendance *Composer*.

Les tests ont pu être validés.

# 6 CONCLUSION

## 6.1 Générale

En conclusion, *Passion Lecture* est une application facile d'utilisation permettant à ses utilisateurs de partager leurs découvertes littéraires en toute simplicité.

Dans la version finale de *Passion Lecture*, toutes les fonctionnalités requises du cahier des charges ont été implémentées. De plus, certaines fonctionnalités optionnelles ont pu être mises en place (notamment *design pattern* MVC, messages d'erreurs qui apparaissent dynamiquement en dessous du bon champ, gestion de la date de sortie plutôt que de l'année), la modification n'a pas pu être complétée à temps (la validation ne se fait pas).

## 6.2 Personnelle

### 6.2.1 Abigaël

Lors de la réalisation de ce projet, je me suis sentie en confiance avec mes camarades. Notre équipe variée en terme de compétences et d'idée a rendu l'ambiance très agréable et l'entraide a été omniprésente pendant l'ensemble du projet. Cependant, j'ai ressenti un inconfort lors de la mise en pratique d'un pattern MVC. Comme c'était un sujet nouveau, j'ai eu de la peine à identifier les différentes vues et comprendre les différents *controller* et leur utilisation. Cette difficulté a affecté ma quantité de code fonctionnelle rendu.

### 6.2.2 Valentin

Ce projet m'a permis de mettre en pratique le modèle MVC avec différents *controller* qui assurent les fonctionnalités demandées. Pour des raisons de manque de temps, certaines sections de code sont dupliquées et n'ont pas pu être refactorisées. En général le projet s'est bien déroulé malgré les contraintes de temps et les nombreux points à corriger en fin de projet.

### 6.2.3 Santiago

Personnellement, je peux conclure de ce travail que j'ai beaucoup appris du travail de mon équipe, car ils ont plus d'expérience dans le domaine et l'analyse de leurs parties du travail m'a permis de comprendre beaucoup de choses pour avancer et être utile.

### 6.2.4 Sébastien

Au début du projet, il m'a fallu un peu d'adaptation. J'avais de l'expérience avec le *design pattern* MVC en Java et en C#, mais jamais en développement web (j'ai particulièrement

été « perturbé » par le fait que le code n'est pas compilé et qu'il a fallu recréer une instance de chaque *controllers* à chaque utilisation).

Autrement, je suis plutôt satisfait de la qualité et de la réutilisabilité du code que j'ai écrit. J'ai modularisé un maximum le code PHP en le séparant en classes et en méthodes, et j'ai pu créer une sorte de *framework* CSS inspiré du fonctionnement de *Tailwind*.

## 6.3 Critique

Lors du développement, nous n'avons quasiment pas touché au rapport, ce qui posa un problème lors des dernières séances du module. En effet, nous avons sous-estimé la quantité d'information à y inscrire et sa rédaction pris beaucoup de temps.

Aussi, certaines parties du code auraient pu être plus modulaire, ce qui aurait facilité la création de tests unitaires.

### 6.3.1 Pistes d'amélioration

Il reste quelques bugs qui, bien que ne posant pas de problèmes majeurs quant à l'utilisation de *Passion Lecture*, auraient pu être corrigés si un peu plus de temps avait été à disposition.

Notamment, les éléments suivants pourraient être améliorés ou corrigés :

- Lors de la déconnexion, l'utilisateur est redirigé sur la page de connexion.
- Plus de tests pourraient être réalisés.
- Le code de validation des formulaires pourrait être refactorisé
- Une image valide lors de l'ajout d'un livre devrait être gardée en mémoire si une autre erreur est envoyée par le formulaire.
- Le formatage « long » des dates (7 January 2025) ne respecte pas la locale de l'utilisateur.

## 7 SOURCES

Les sources ont été citées en notes de bas de page ou, le cas échéant, directement dans le code source.

## 8 GLOSSAIRE

Plateformes existantes: Des sites déjà établis et bien connus, qui servent de modèle pour d'autres plateformes, comme Goodreads et Anilist, qui sont des plateformes pour les livres et les animes.

Hash: Un processus cryptographique qui transforme un mot de passe en une chaîne de caractères fixe et sécurisée, utilisée pour stocker le mot de passe de manière sécurisée dans la base de données.

Modèle: Une représentation ou une structure qui montre comment les données sont organisées et interconnectées dans un système.

Entité: Un élément ou une unité dans une base de données, comme un auteur, un livre, un éditeur, une catégorie, ou un utilisateur, qui possède des propriétés et peut interagir avec d'autres entités.

MCD (Modèle Conceptuel de Données): Un modèle qui décrit les entités d'un système et leurs relations à un niveau conceptuel.

MLD (Modèle Logique de Données): Un modèle qui détaille la manière dont les données sont organisées dans une base de données spécifique en termes de tables, de colonnes, et de relations.

Pattern 7-1: Méthodologie pour structurer les fichiers SCSS, divisant le code en 7 dossiers principaux (abstracts, base, components, layout, pages, thèmes, vendors) et 1 fichier principal pour les importer.

Écoconception (conception écologique): Pratiques visant à réduire l'impact écologique d'une application, comme l'utilisation de la pagination plutôt que du défilement infini pour améliorer l'efficacité et limiter la consommation de ressources.

Filtres "filter sanitize" de PHP: Fonctions PHP qui appliquent des filtres pour assainir (sanitizer) les données entrées par l'utilisateur en les transformant en un format sûr. Par exemple, les caractères spéciaux comme <, >, &, etc., sont convertis en leurs équivalents d'entités HTML pour éviter les attaques XSS (Cross-Site Scripting).

Entités HTML: Représentation des caractères spéciaux sous forme de codes HTML, comme &lt; pour < ou &gt; pour >. Cela permet de neutraliser les caractères pouvant être utilisés dans des attaques XSS ou des injections de code malveillant.

RFC 2396: La norme (Request for Comments) qui définit le format d'une URL, incluant la syntaxe et les caractères autorisés dans une URL. Le contrôle basé sur cette norme s'assure que l'URL respectera le format attendu pour être valide.

Regex (Expressions régulières): Les expressions régulières (ou regex, de l'anglais "regular expressions") sont des séquences de caractères qui forment un motif de recherche. Elles

permettent de rechercher, de valider ou de manipuler des chaînes de texte selon des règles spécifiques. Par exemple, une regex peut être utilisée pour vérifier si une adresse e-mail est valide, ou pour extraire des informations précises d'un texte.

[ISO 8601](#) : Le format ISO 8601 est une norme internationale pour représenter des dates sous la forme YYYY-MM-DD, par exemple "2025-01-07". La classe FormatHelper transforme cette chaîne de caractères en un format plus lisible pour les utilisateurs, par exemple "7 janvier 2025"

[PHPUnit](#) : Outil de test unitaire pour PHP. Il permet de réaliser des tests automatisés pour s'assurer que le code fonctionne comme prévu. PHPUnit est souvent utilisé dans le cadre de l'intégration continue et de l'amélioration de la qualité du code.