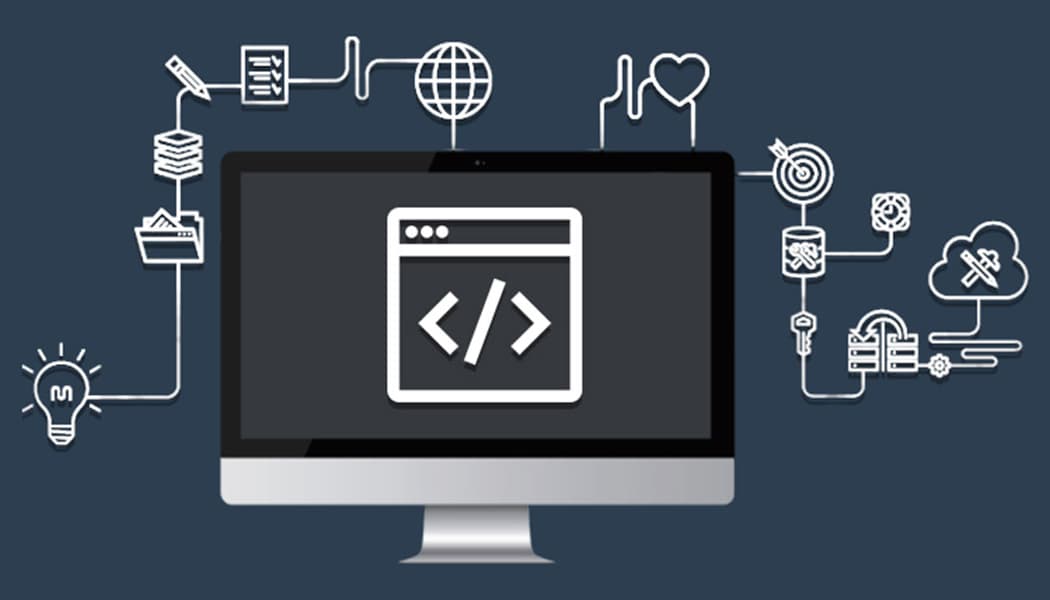
**P\_WEB295**



(Logo Backend)

Antoine Piguet  
Yosef Nademo  
Charles-Henri Moser  
 CID2B

Vennes - Lausanne

24 périodes

Chef de Projet – Gregory Charmier

Table des matières

[1 Introduction 3](#_Toc195003251)

[2 Analyse 4](#_Toc195003252)

[2.1 Planification – GitHub Project 4](#_Toc195003253)

[2.2 Routes – API REST 5](#_Toc195003254)

[2.3 Conception de la base de données 7](#_Toc195003255)

[3 Réalisation 8](#_Toc195003256)

[3.1 Algorithmes utilisés 8](#_Toc195003257)

[3.2 Maquettes – Figma 8](#_Toc195003258)

[3.3 Démarche d’éco-conception et Responsabilité Numérique 8](#_Toc195003259)

[4 Tests 9](#_Toc195003260)

[4.1 Dossier des tests 9](#_Toc195003261)

[5 Conclusion 9](#_Toc195003262)

[5.1 Conclusion générale 9](#_Toc195003263)

[5.2 Bilan personnel 9](#_Toc195003264)

[5.2.1 Conclusion Charles-Henri 9](#_Toc195003265)

[5.2.1 Conclusion Antoine Piguet 9](#_Toc195003266)

[5.2.2 Conclusion Yosef Nademo 9](#_Toc195003267)

[6 Divers 10](#_Toc195003268)

[6.1 Webographie/ Bibliographie 10](#_Toc195003269)

[7 Annexes 10](#_Toc195003270)

# Introduction

Le projet Passion Lecture a pour but de dévelpper un backend d'une application dédiée aux amateurs de lecture. Le projet est intégré au module 295 et devrait permettre aux utilisateurs de partager, voir et évaluer les livres. L'application sera basée sur une API REST sécurisée qui devrait permettre de gérer les livres, les catégories, les utilisateurs et les relations entre eux.

L’API devra intégrer une authentification sécurisée avec JWT, une validation rigoureuse des données, ainsi qu’une gestion des erreurs avec des statuts HTTP appropriés. Pour une expérience utilisateur fluide, une documentation détaillée sera générée via Swagger, et des tests seront réalisés avec Insomnia, Postman et des tests automatisés avec Vitest.

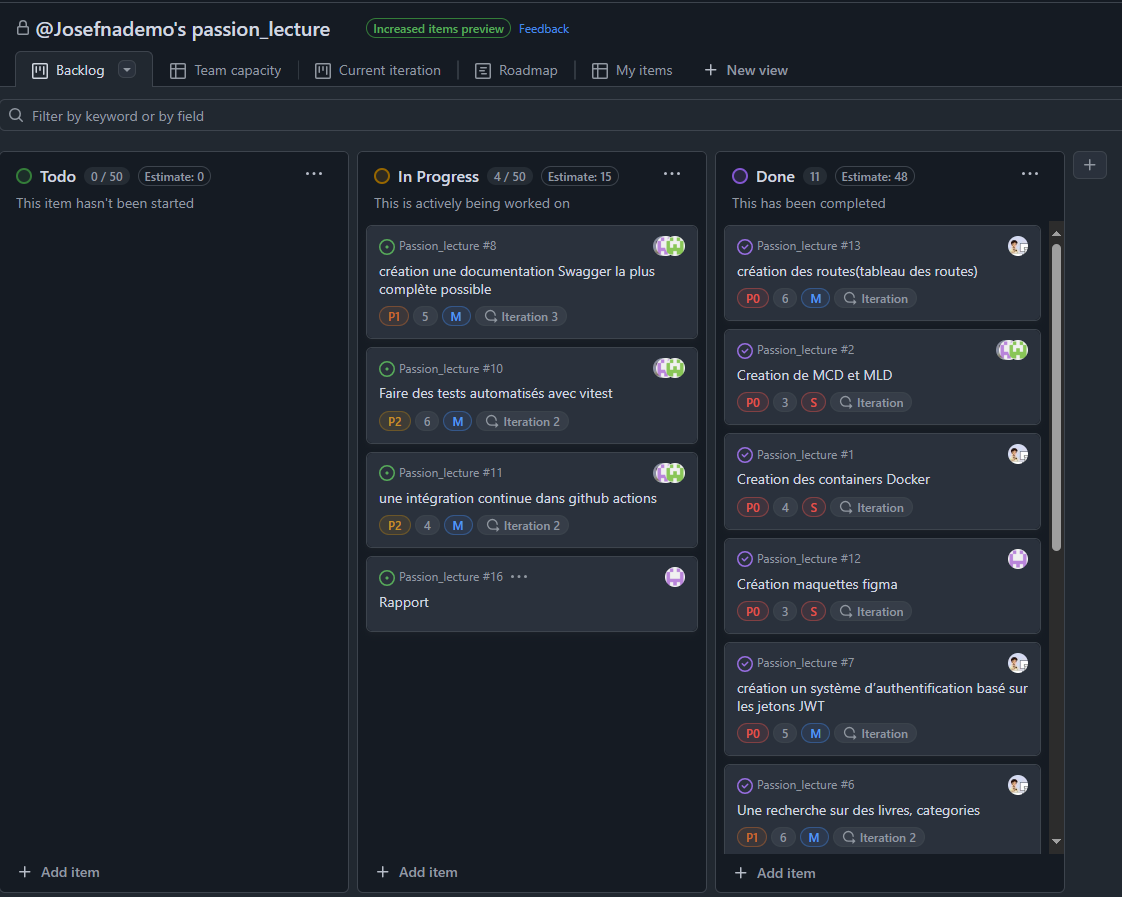
Le back-end sera développé avec Node.js, Express.js et Sequelize pour gérer une base de données MySQL. Le projet utilisera l'intégration continue avec GitHub Actions et la conteneurisation avec Docker pour assurer la portabilité. Finalement, le design écologique sera pris en compte, afin de proposer une application optimisée et respectueuse des normes de développement durable.

# Analyse

## Planification – GitHub Project

Durant le projet, nous avons utilisé GitHub Project afin d’organiser nos tâches et de se les répartir afin d’avoir une exécution plus efficace de notre projet.

Comme vous pouvez le voir sur cette image (ci-dessous), Nous avons utilisé le modèle de tableau Kanban avec trois colonnes : le backlog, c’est ici que nous avons mis toutes les tâches que nous devions accomplir, la colonne -In progress- ou nous mettons les tâches que nous sommes actuellement entrain de faire, et la colonne -Done-, ici nous updatons nos tâches afin de les définir comme finis AKA Done.



(Image tableau Kanban)

## Routes – API REST

|  |  |  |  |
| --- | --- | --- | --- |
| Action | Chemin | Json ? | Description |
| POST | /books/ | Toute la t\_livre + image | Ajoute un livre. |
| DELETE | /books/ :id/ | Non | Supprimer un livre. |
| PUT | /books/ :id/ | Toute la t\_livre | Mettre à jour un livre. |
| DELETE | /books/:id/notes/:id/ | Non | Supprimer un commentaire et une note par ID |
| PUT | /books/:id/notes/:id/ | La note + commentaire + UserID | Mettre à jour un commentaire et une note par ID |
| GET | /books/:id/notes/:id/ | Non | Obtenir une évaluation par son ID |
| POST | /books/:id/notes/ | La note + commentaire + UserID | Poste une note |
| GET | /books/:id/notes/ | Non | Regarde toutes les notes d’un livre |
| GET | /books/ :id/ | Non | Récupère un ivre par id |
| GET | /books/ | Non | Cherche un livre avec les paramètres demandés. |
| GET | /authors/ | Non | Récupère tous les auteurs |
| GET | /authors/:id/ | Non | Récupère les auteurs correspondant à un ID |
| GET | /authors/:id/books/ | Non | Récupère tous les livres d’un auteur |
| GET | /categories/ | Non | Récupère toutes les catégories |
| GET | /categories/:id/books/ | Non | Récupère tout livres qui correspond à une catégorie |
| GET | /categories/:id/ | Non | Récupère les categories qui correspondent à l’id |
| GET | /users/ | Non | Récupère tous les user |
| DELETE | /users/:id | Non | Supprimer un user par son ID |
| POST | /users/ | username + password | Ajouter un user |
| POST | /auth/login/ | username + password | Se connecter |

## Conception de la base de données

Le modèle relationnel de données a été conçu avec MySQL et Sequelize :

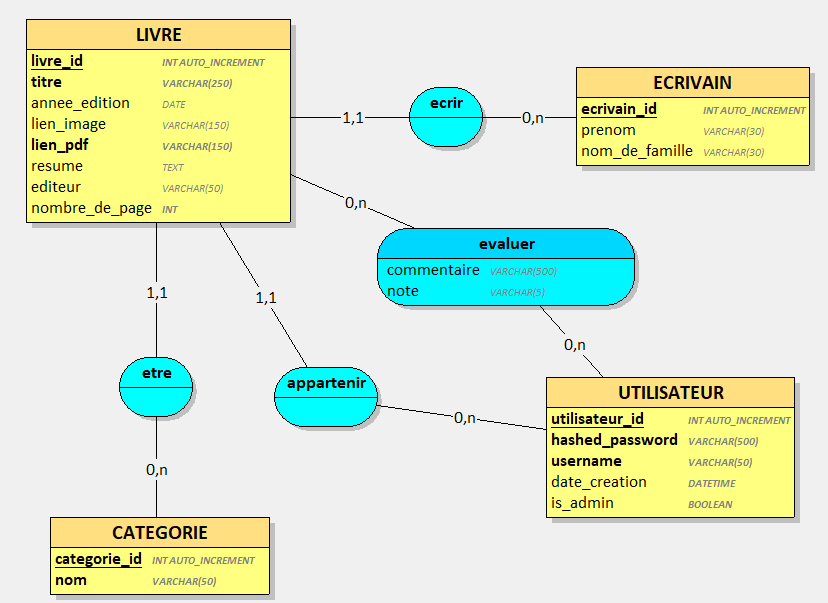
Tables principales : Utilisateur, Livre, Categories, Ecrivain

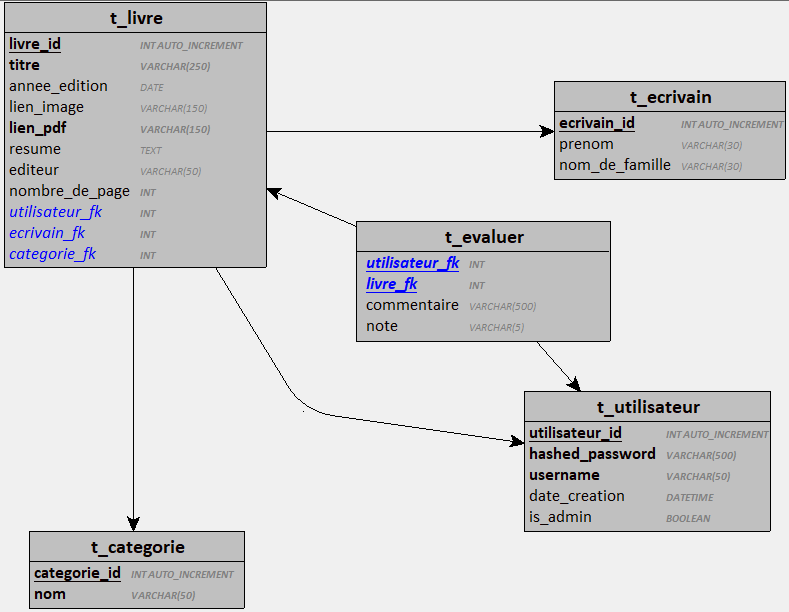
Relations :

Un utilisateur peut ajouter plusieurs livres

Un livre appartient à une catégorie et a un écrivain

Un utilisateur peut noter et commenter plusieurs livres





# Réalisation

## Algorithmes utilisés

L'authentification repose sur JWT (JSON Web Token) pour assurer une gestion sécurisée des accès utilisateur

## Maquettes – Figma

Pour visualiser la maquette de notre projet, vous pouvez consulter notre prototype interactif disponible sur Figma en cliquant sur le lien ci-dessous : [Maquettes PDF](Figma/Passion-Livres.pdf)

## Démarche d’éco-conception et Responsabilité Numérique

Dans un contexte où le numérique a un impact environnemental croissant, ce projet s’inscrit dans une **démarche d’éco-conception** visant à minimiser son empreinte écologique tout en garantissant la qualité et la performance.

Plusieurs principes ont été intégrés :

* **Optimisation des ressources** : le code backend a été structuré de manière à éviter les calculs redondants, limiter les appels inutiles à la base de données et favoriser la réutilisation logique des composants.
* **Réduction des transferts de données** : en structurant efficacement les requêtes et en limitant les réponses aux informations essentielles, la bande passante est préservée, ce qui réduit la consommation énergétique.
* **Utilisation raisonnée des dépendances** : seules les bibliothèques strictement nécessaires ont été intégrées, évitant ainsi un alourdissement inutile du projet.
* **Tests légers et ciblés** : réalisés avec *Insomnia*, les tests manuels ont été effectués de manière locale, évitant les déploiements superflus et limitant l’utilisation des ressources cloud.
* **Documentation claire (Swagger)** : une documentation bien structurée assure une meilleure maintenabilité du projet, prolongeant ainsi son cycle de vie et réduisant la dette technique.
* **Responsabilité collective** : l’équipe projet (Antoine Piguet, Yosef Nademo, Charles-Henri Moser) a été sensibilisée à l’importance d’une approche responsable dans le développement, guidée par le chef de projet Grégory Charmier.

# Tests

## Dossier des tests

Les tests ont été réalisés à l'aide de l'outil **Insomnia**, permettant d'effectuer des tests manuels sur les différentes routes de l’API.  
Une **vidéo démonstrative** est également fournie, attestant du bon fonctionnement de l’ensemble des routes testées.

# Conclusion

## Conclusion générale

Mon avons pus conclure la plupart des fonctionnalités demandée mais il manque les tests avec Vitest car nous n’avons pas eu le temps.

## Bilan personnel

### Conclusion Charles-Henri

En conclusion, j’ai beaucoup apprécié le projet, car il m’a permis d’en apprendre énormément sur le backend et comment est-ce qu’il fonctionne, comme la structure et les différentes parties du code (modèle, middleware, Controller, routes, …) Et ceci est extrêmement propice pour notre futur en tant que développeurs. Car après se projet, nous allons faire le frontend, ce qui veux dire que nous faisons la réalisation complète du frontend e du backend. Si c’étais à refaire, je pense que de séparer le frontend et le backend en deux trimestres différent pourrait être bénéfique à la meilleure compréhension totale du projet. Mais finalement, j’ai beaucoup appris et apprécié ce projet backend.

### Conclusion Antoine Piguet

J’ai aussi apprécié le projet car j’ai appris plein de choses entre autres sur l’utilisation des ORM et de plusieurs fonctionnalités de JS. J’ai l’impression d’avoir bien progressé et d’avoir acquis des connaissances.

Il faudra néanmoins que l’on apprenne à mieux communiquer en équipe je pense car nous avons eus quelques soucis pour finir pour la deadline car Yosef n’a pas update de tout le weekend alors que le code et le rapport n’étaient pas terminés, c’est pourquoi il n’a pas de conclusion.

### Conclusion Yosef Nademo

Ce projet m'a permis de développer mes compétences en backend, en particulier avec Node.js, Express.js, et JWT pour l'authentification. J'ai aussi appris l'importance de la communication et de la gestion du temps au sein de l'équipe. Bien que la majorité des fonctionnalités aient été réalisées, des retards sur les tests avec Vitest ont retardé la fin du projet. Si c'était à refaire, je m'assurerais d'une meilleure coordination pour respecter les délais. Ce projet m'a préparé à la phase frontend et a enrichi ma compréhension du développement complet d'une application.

# Divers

## Webographie/ Bibliographie

Documentation Sequelize : <https://sequelize.org/docs/v6/>

Documentation Express.js : <https://expressjs.com/>

Swagger API Docs : <https://swagger.io/docs/>

Tutoriels JWT et sécurité des API : <https://jwt.io/introduction/>

# Annexes

Repo GitHub: <https://github.com/Josefnademo/Passion_lecture.git>

Figma : [Passion-Livres – Figma](https://www.figma.com/design/iG5BIscos6hSl4IACsA2Ek/Passion-Livres?node-id=0-1&p=f&t=zx3GCyWmoT17IoEr-0)

GitHub Project : [Backlog · @Josefnademo's passion\_lecture](https://github.com/users/Josefnademo/projects/2/views/1)