P\_WEB – C295



Sofien Habib Belkhiria - Lucas Lordon

Vennes – ETML

19.02.24 – 12.03.24

Chef de projet : Antoine Mveng

Table des matières

[Introduction 1](#_Toc157756702)

[Conclusion 1](#_Toc157756703)

# Introduction

Ce projet à été réaliser par Lucas Lordon et Sofien Habib Belkhiria.

Le but de ce projet est de faire une application backend qui consiste à la mise en place d’une API REST. L’API REST doit reprendre les notions suivantes vu dans le cours : les routes, la gestion des statuts, un système de recherche, un système d’authentification JWT, une documentation Swagger, des tests via Insomnia, des tests automatisés avec vitest, GitHub actions et une dockerisation du backend.

Le développement de cette application permettra de consolider les bases de nos connaissances sur le développement Backend et l’utilisation de technologies tels que « Sequilize » et « Insomnia ».

Le projet à été réaliser entre le 19.02.24 et le 12.03.24 pour un total de 24 périodes.

L’ensemble du projet et présent dans le repository GitHub P\_Web295, mis à par la planification qui a été réaliser sur trello.

To Do : Comprend une brève explication du projet (½ page)2

# Analyse

To Do : Planification de toutes les tâches à réaliser : Vous devez utiliser un outil permettant au chef de projet de voir l’évolution des tâches (github projet ou trelllo). Contiendra une analyse de l’API REST présentant pour chaque route : Le verbe http, l’URI et si nécessaire le json ou autre information à transmettre (1 page) Contiendra une analyse de de la base de données à réaliser (MCD, MLD, MPD) (1 page) Contiendra une analyse de la structure du code qui sera effectuée (Organisation du code, …) (1 page) Schéma de l’architecture représentant les interactions entre le frontend et le backend tout en situant les différents composants (API REST, la base de données, l’ORM, etc)

## Routes

Livre

|  |  |  |  |
| --- | --- | --- | --- |
| Verbes | Utilisation | Exemple URI | JSON |
| Get | Retourne toutes les données d’une table. | http://localhost/api/books |  |
| Get/id | Retourne un champ précis de la db à condition que la personne rentre l’identifiant du champ. | http://localhost/api/books/2 |  |
| POST | Permet d’ajouter un champ à la DB tant que l’ajout respecte la structure de la DB. |  |  |
| DELETE /id | Permet de supprimer un champ précis de la DB à condition de connaitre son identifiant. |  |  |
| PUT/id | Permet de mettre à jour une entrée de la DB à condition de connaitre son identifiant. |  |  |

# Réalisation

# Test

# Conclusion

# Webographie/Bibliographie/Glossaire