P\_app\_183



Antoine Fabre – CID2B

ETML - Vennes

8 semaines

13.01.2025 – 10.03.2025

M. Schaffter

Table des matières

[1 Spécifications 3](#_Toc188612895)

[1.1 Titre 3](#_Toc188612896)

[1.2 Description 3](#_Toc188612897)

[1.3 Matériel et logiciels à disposition 3](#_Toc188612898)

[1.4 Prérequis 3](#_Toc188612899)

[1.5 Cahier des charges 3](#_Toc188612900)

[1.5.1 Objectifs et portée du projet (objectifs SMART) 3](#_Toc188612901)

[1.5.2 Caractéristiques des utilisateurs et impacts 3](#_Toc188612902)

[1.5.3 Fonctionnalités requises (du point de vue de l’utilisateur) 3](#_Toc188612903)

[1.5.4 Contraintes 3](#_Toc188612904)

[1.5.5 Travail à réaliser par l'apprenti 3](#_Toc188612905)

[1.5.6 Si le temps le permet … 4](#_Toc188612906)

[1.5.7 Méthodes de validation des solutions 4](#_Toc188612907)

[1.6 Les points suivants seront évalués 4](#_Toc188612908)

[1.7 Validation et conditions de réussite 4](#_Toc188612909)

[2 Planification Initiale 4](#_Toc188612910)

[3 Analyse 4](#_Toc188612911)

[3.1 Opportunités 4](#_Toc188612912)

[3.2 Document d’analyse et conception 4](#_Toc188612913)

[3.3 Conception des tests 4](#_Toc188612914)

[3.4 Planification détaillée 5](#_Toc188612915)

[4 Réalisation 5](#_Toc188612916)

[4.1 Dossier de Réalisation 5](#_Toc188612917)

[4.1.1 Script de création de la base de données 5](#_Toc188612918)

[4.1.2 Page de login 6](#_Toc188612919)

[4.1.3 Page d’enregistrement 6](#_Toc188612920)

[4.2 Modifications 7](#_Toc188612921)

[5 Tests 7](#_Toc188612922)

[5.1 Dossier des tests 7](#_Toc188612923)

[6 Conclusion 7](#_Toc188612924)

[6.1 Bilan des fonctionnalités demandées 7](#_Toc188612925)

[6.2 Bilan de la planification 7](#_Toc188612926)

[6.3 Bilan personnel 7](#_Toc188612927)

[7 Divers 7](#_Toc188612928)

[7.1 Journal de travail 7](#_Toc188612929)

[7.2 Bibliographie 7](#_Toc188612930)

[7.3 Webographie 7](#_Toc188612931)

[8 Annexes 7](#_Toc188612932)

# Spécifications

## Titre

Création d’un site d’e-commerce sécurisé (webstore)

## Description

Construire une application [node.js](https://nodejs.org/fr) offrant un accès sécurisé et une gestion des rôles.

## Matériel et logiciels à disposition

* Un ordinateur standard de la section informatique
* Docker Desktop
* Node.js
* Visual Studio Code

## Prérequis

Première année de formation à l’ETML

## Cahier des charges

### Dockerisation

L’ensemble des services web sera conteneurisé.

### Profil du client

Le client peut accéder à son propre profil depuis la page de bienvenue. Seul son profil lui sera rendu visible et les routes du backend seront protégées.

### HTTPS

Il doit être possible d’accéder à votre site de e-commerce de manière sécurisée (https://localhost). Le port utilisé sera le 443. Le certificat sera auto-signé par OpenSSL.

### Authentification par mot de passe

L’utilisateur devra s’authentifier par la page https://localhost/login Le mot de passe sera haché et salé avant d’être stocké dans la base de données (table t\_users).

### Vérification du token JWT

Le token devra être vérifié lorsque l’utilisateur vous l’envoie et la signature du jeton devra aussi être utilisée pour vérifier que le token n’a pas été modifié.

### Administration

Une page d’administration devra avoir un champ de recherche (Nom du visiteur) et permettre d’afficher tous les utilisateurs ayant tout ou partie de ce nom.

### Protection contre les injections SQL

Votre page d’administration devra être protégée contre les injections SQL. Sans utiliser sequelize ou tout autre ORM, votre site devra être robuste face aux injections. L’utilisation de sequelize (ou autre ORM) sera autorisée dans l’approfondissement du projet.

### Utilisation de bcrypt

Une fois l’authentification faite avec votre propre mécanisme (mot de passe en base de données, salé et haché). Vous modifierez votre code pour utiliser bcrypt. Votre code doit prévoir une condition pour utiliser soit bcrypt soit votre propre implémentation.

### Versioning

Votre code sera versionné sur Github et un .gitignore empêchera de versionner les binaires npm (dossiers node\_modules, documents Word/Excel). Votre dépôt sera partagé avec votre chef de projet dès le début du projet.

### Documentation

Journal de travail (1 ligne par quart d’heure de travail)

• Rapport, contenant :

o Votre conceptualisation (schéma)

o Des explications sur votre code

o Une conclusion sur le travail fourni et sur l’attitude face au projet

### Utilisation de l’IA

Une utilisation injustifiée de l’IA ou une absence de maîtrise du code, induira un non-acquis dans les compétences techniques du projet.

## Les points suivants seront évalués

* Le rapport
* Les planifications (initiale et détaillée)
* Le journal de travail
* Le code et les commentaires
* Les documentations de mise en œuvre et d’utilisation

## Validation et conditions de réussite

* Compréhension du travail
* Possibilité de transmettre le travail à une personne extérieure pour le terminer, le corriger ou le compléter
* Etat de fonctionnement du produit livré

# Planification Initiale

La planification se trouve dans le fichier Excel « *T-P\_app\_183-ThodeMateo-planification\_jdt.xlsm* ».

# Analyse

## Opportunités

Pour mener à bien ce projet il faut acquérir ces compétences :

* Comment créer et vérifier des token JWT
* Savoir crypter un mot de passe avec crypto
* Comment accéder aux cookies et mettre des informations dans les cookies
* Gérer des utilisateurs et les rôles
* Comment mettre un site en https
* Comment se protéger des injections SQL

## Document d’analyse et conception

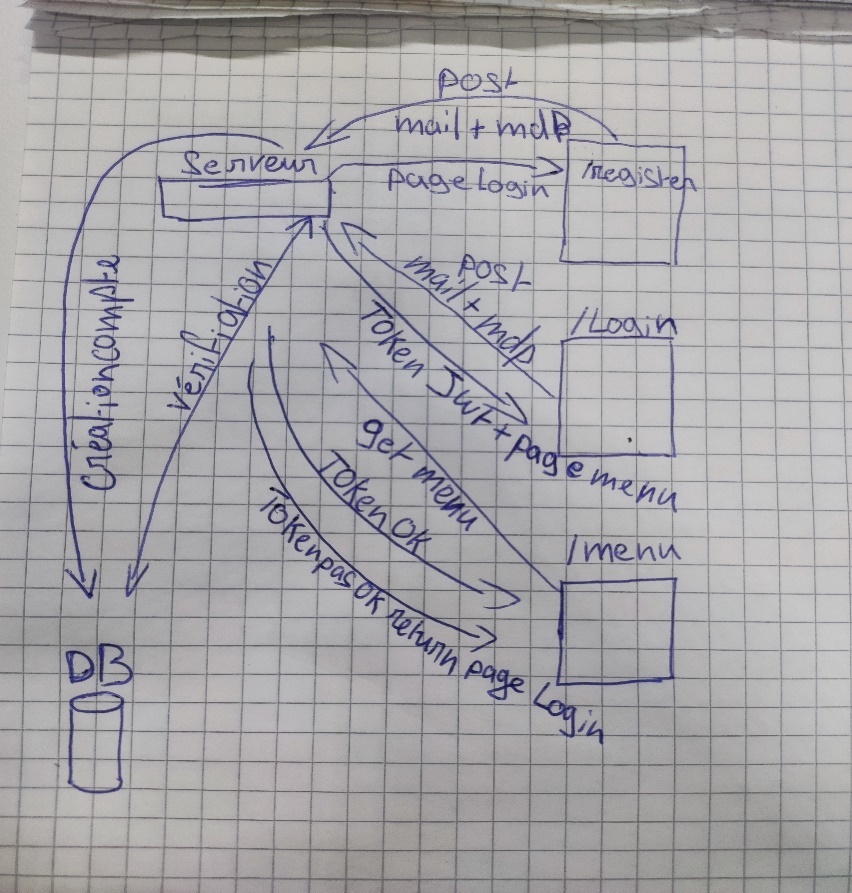


Figure schéma site Web

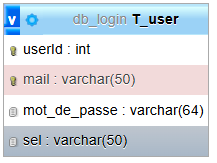
Sur ce schéma on voit les trois routes qu’on a besoin pour ce site web.

Donc la première route c’est la route « /register » avec laquelle on peut créer un compte en envoyant un mail et un mot de passe au serveur qui va créer un sel, qui va hacher le mot de passe pour mettre dans la base de données. Un fois que c’est fait le serveur redirige l’utilisateur vers la page de login.

La deuxième route est la route « /login » sur laquelle l’utilisateur peut se connecter en envoyant le mail et le mot de passe au serveur qui lui va récupérer les infos dans la base de données, hacher le mot de passe donné par l’utilisateur et le serveur regarde si c’est le bon mot de passe. Si c’est le bon mot de passe le serveur renvoi à l’utilisateur un token JWT qui sera stocké dans les cookies et redirige l’utilisateur vers la route « /menu ».

La troisième route est la route « /menu » cette route sert à afficher le menu seulement si l’utilisateur est connecté. Et pour savoir si l’utilisateur est déjà connecté il faut que l’utilisateur ait un token JWT dans ses cookies et comme ça le serveur vérifie le token. Si le token est valide le serveur envoi la page menu et si le token n’est pas fourni ou alors le token n’est pas valable le serveur renvoi l’utilisateur a la page login pour qu’il se connecte.

Et dans ce projet on a besoin d’une base de données pour stocker le mail, le mot de passe hacher et le sel des utilisateurs. Donc voici le schéma de la base de données.



## Planification détaillée

A ce stade, après l’analyse complète du projet, un planning détaillé et complet (avec tâches, sous-tâches, dépendances, durée, …) peut être finalisé.

Le planning détaillé doit s’inscrire dans le planning initial. Il faut que l’on puisse situer cette planification détaillée par rapport à la planification initiale.

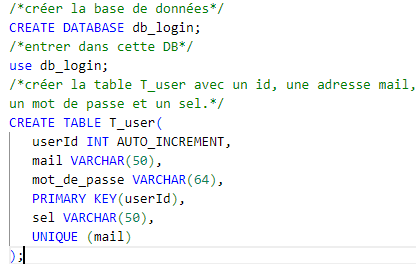
# Réalisation

## Dossier de Réalisation

### Script de création de la base de données

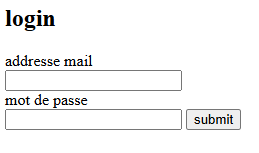
On commence par créer la base de données et on créer la table t\_user.

Avec ce script on peut recréer la base de données autant de fois qu’on veut s’il y a un problème avec PHPMyAdmin lors de la phase de développement.

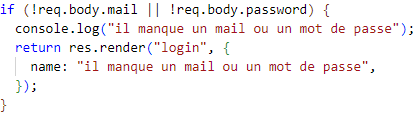


### Page de login

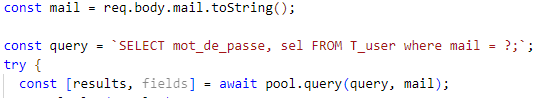
Voici la page de login que j’ai créée :



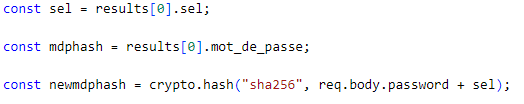
On peut donc se connecter en entrant son adresse électronique et son mot de passe. Quand on se connecte on appelle la route POST /login qui va regarder déjà s’il y a un mot de passe et un mail.



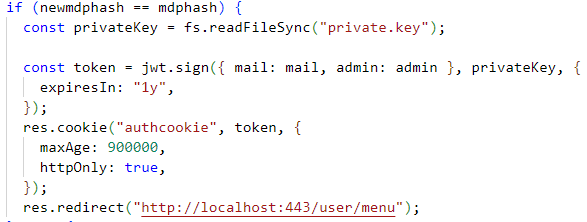
Ensuite on prend le mail et on fait une requête a la base de données pour récupérer le mot de passe hacher, le sel et admin (admin est un booléen, donc pour savoir si l’utilisateur est admin).



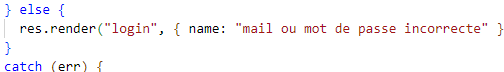
Donc après qu’on a récupéré le mot de passe haché, le sel et admin, on hache le mot de passe donné par l’utilisateur avec le sel récupérer juste avant.



Ensuite on regarde si le mot de passe envoyer par l’utilisateur et le mot de passe dans la base de données sont les mêmes. Si ce sont les mêmes on récupère la clé privée et avec elle, on créé un token JWT qui dure 1 an. Dans ce token on met dans le Payload le mail de l’utilisateur et s’il est admin. Ce token est envoyer dans les cookies et l’utilisateur est redirigé vers le menu.



Si les 2 mots de passe ne sont pas les même alors on renvoie l’utilisateur a la page login avec un message.



### Page de register



Sur cette dernière on peut entrer une adresse mail et un mot de passe.

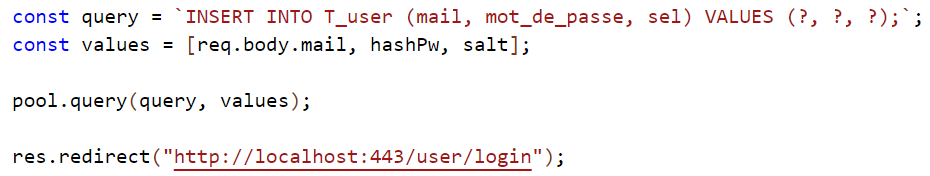
Si on appuie sur le bouton « submit » on appelle la route POST /register qui va en premier créer un sel pour cet utilisateur.



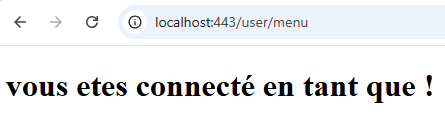
Ensuite ça hache le mot de passe grâce au sel créé juste avant.



Et pour finir on insert les données dans la base de données et on redirige l’utilisateur vers la route login.

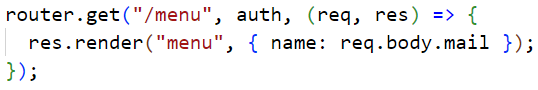


### Page de menu

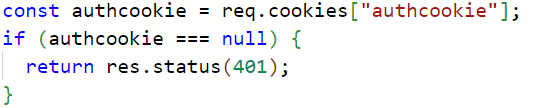


Pour avoir accès a cette page il faut en premier se connecter donc passer par la route login. Mais si quelqu’un veut accéder à cette page.

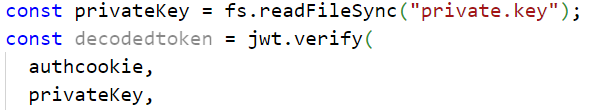
Il va devoir passer par la méthode « auth ».



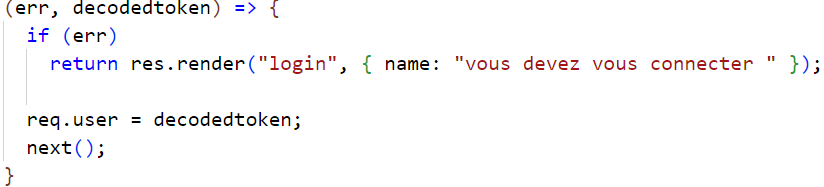
Cette méthode va en premier aller dans les cookies et regarder s’il y a un token JWT.



Ensuite ça récupère la clé privée utilisé pour faire le token, pour vérifier que le token est valide.



Si le token n’est pas ou plus valide on revoit l’utilisateur la page de login avec un message d’erreur. Sinon on passe à la suite (donc afficher la page de menu).



### Page d’admin



Cette page est la page d’admin, pour y avoir accès

# Conclusion

## Bilan des fonctionnalités demandées

Il s’agit de reprendre point par point les fonctionnalités décrites dans les spécifications de départ et de définir si elles sont atteintes ou pas, et pourquoi.

Si ce n’est pas le cas, estimer en « % » ou en « temps supplémentaire » le travail qu’il reste à accomplir pour terminer le tout.

## Bilan de la planification

Distinguer et expliquer les tâches qui ont généré des retards ou de l'avance dans la gestion du projet. Indiquer les différence entre les planifications initiales et détaillées avec le journal de travail.

## Bilan personnel

Si c’était à refaire:

Qu’est-ce qu’il faudrait garder ? Les plus et les moins ?

Qu’est-ce qu’il faudrait gérer, réaliser ou traiter différemment ?

Qu’est que ce projet m’a appris ?

Suite à donner, améliorations souhaitables, …

Remerciements, signature, etc.

# Divers

## Journal de travail

Le journal de travail se trouve dans le dossier « livrables », il est nommé « T-P\_app\_183-ThodeMateo-planification\_jdt.xlsm » mais n’est **pas** disponible sur GitHub (selon ce qui a été demandé). Il se trouve donc sur [Teams](https://eduvaud.sharepoint.com/:f:/r/sites/ETML_INF-GRP2D-24-25_Teams/Documents%20partages/P_183_SecureWebshop-CSR/Mateo%20Thode?csf=1&web=1&e=K4wqLk), ainsi que le rapport.

## Webographie