P\_APP



Muthulingam Abiram – CID2A

ETML - Vennes

13.01.2025 – 14.03.2025

Table des matières

[1 Spécifications 3](#_Toc192883052)

[1.1 Titre 3](#_Toc192883053)

[1.2 Description 3](#_Toc192883054)

[1.3 Matériel et logiciels à disposition 3](#_Toc192883055)

[2 Votre conceptualisation (schéma) 4](#_Toc192883056)

[3 Réalisation 5](#_Toc192883057)

[3.1 HTTPS : 5](#_Toc192883058)

[3.2 Profil du client : 7](#_Toc192883059)

[3.3 Authentification par mot de passe : 7](#_Toc192883060)

[3.4 Vérification du token JWT : 8](#_Toc192883061)

[3.5 Administration : 8](#_Toc192883062)

[3.6 Protection contre les injections SQL : 9](#_Toc192883063)

[3.7 Utilisation de bcrypt : 9](#_Toc192883064)

[4 Conclusion 9](#_Toc192883065)

# Spécifications

## Titre

P\_APP : Création d’un site d’e-commerce sécurisé (webstore)

## Description

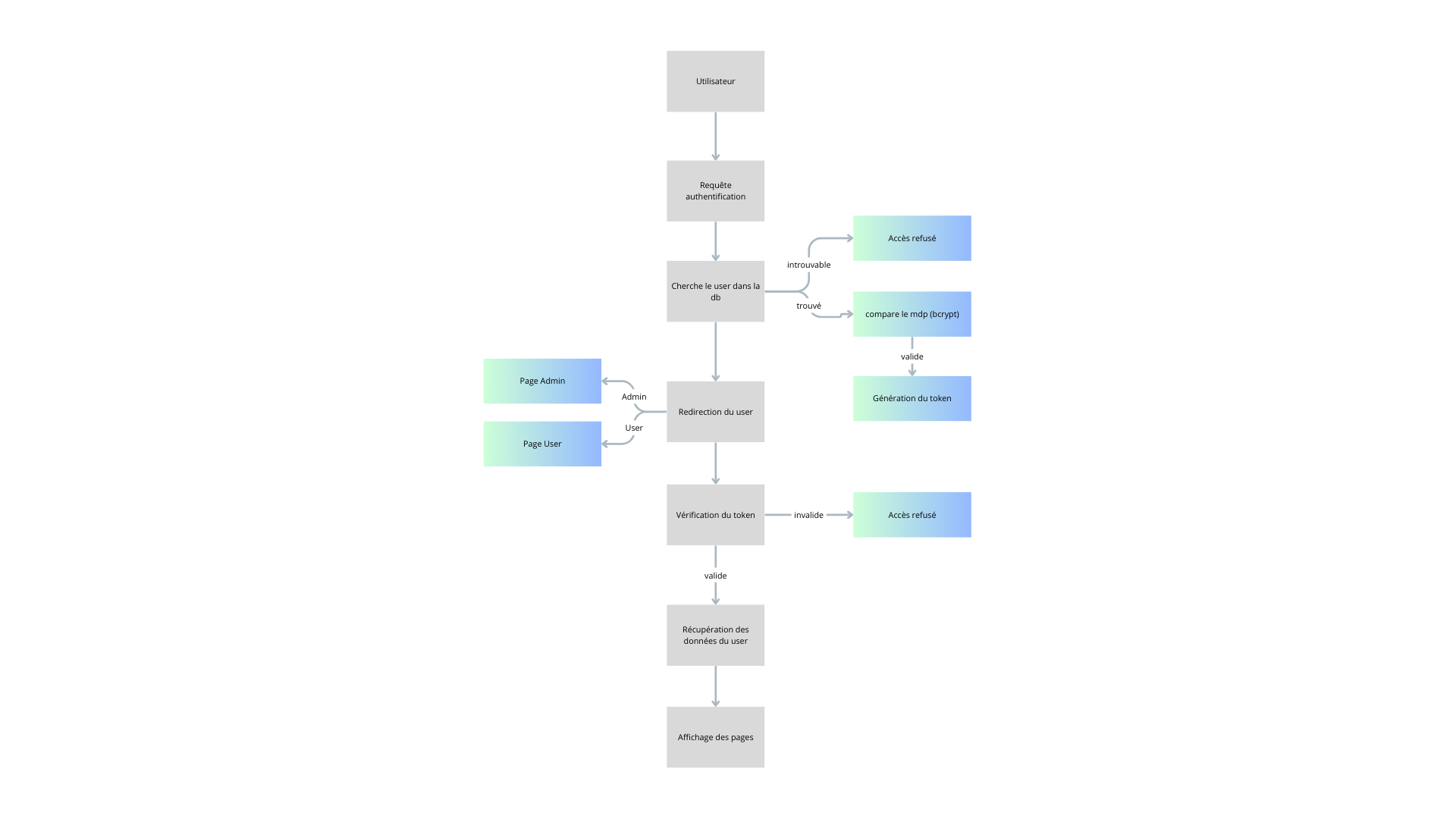
L’objectif de ce projet et de réaliser une application e-commerce sécurisé, qui se concentre sur la page d’authentification (des clients et des administrateurs).

Durant ce projet, il faudra rendre un rapport expliquant le code et le processus du développement de l’application, ainsi que le code permettant de voir l’application et de s’y connecter.

## Matériel et logiciels à disposition

* Internet
* Module I183
* Node.js
* Visual studio

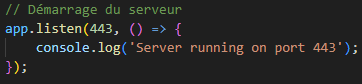
# Votre conceptualisation (schéma)



# Réalisation

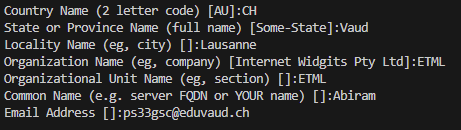
## HTTPS :

Changement du port, qui sera le 443.



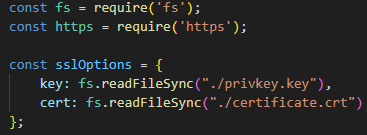
Génération du certificat ssl

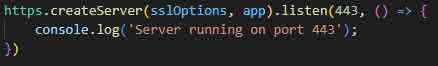
1. openssl genpkey -algorithm RSA -out privkey.key
2. openssl req -new -key privkey.key -out request.csr



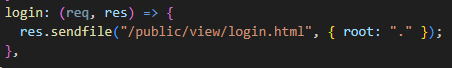
1. openssl x509 -req -in request.csr -signkey privkey.key -out certificate.crt -days 365

Changement http en https :



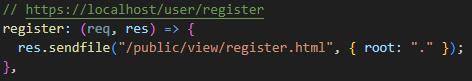


Lien page login depuis le controlleur

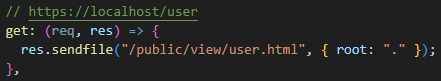


Définis le chemin où se trouve le fichier html, puis root signifie la racine du projet. Le ‘.’ Signifie l’endroit où on se trouve.

Lien page register depuis le controlleur



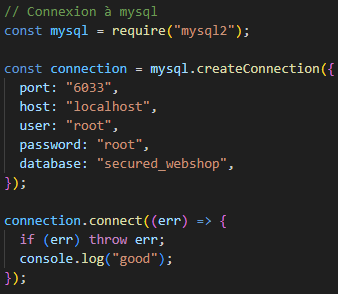
Lien page user depuis le controlleur



Création de la base de données dans phpMyAdmin



Connexion à mysql :



## Profil du client :

La page du profil du client, est une page qui affiche les données de l’utilisateur qui est connecté. Cette page affiche son email, ainsi que son rôle.

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

Une image contenant texte, Police, capture d’écran, ligne

Le contenu généré par l’IA peut être incorrect.

La requête fetch permet de récupérer les informations de l’utilisateur, si seulement l’utilisateur est authentifié. Si l’utilisateur veut retourner à la page d’accueil, il sera redirigé en fonction de son rôle (admin = 1, user = 0).

## Authentification par mot de passe :

L’authentification, se fait par email et un mot de passe (qui est chiffré dans la db).

Une image contenant texte, capture d’écran, affichage

Le contenu généré par l’IA peut être incorrect.

Quand un utilisateur tente de se connecter, le serveur cherche l’utilisateur dans la db à l’aide de la commande ‘SELECT \* FROM t\_users Where email = ?’, qui permet de trouver l’utilisateur correspondant à l’email. Une fois l’utilisateur trouvé, il compare le mot de passe avec bcrypt (mot de passe entré par l’utilisateur et mot de passe hashé dans la db). Si le mot de passe est correct, un token sera généré et s’expirera dans 1h.

## Vérification du token JWT :

Le token JWT, permet de garder la session de l’utilisateur après la connexion.

Une image contenant texte, capture d’écran, affichage

Le contenu généré par l’IA peut être incorrect.

La variable token, récupère le token dans les cookies, si le token n’existe pas, l’accès pour l’utilisateur lui est refusé. Sinon le token est vérifé, s’il est valide ou expiré.

## Administration :

La page pour les administrateurs, permet d’afficher la liste de tous les utilisateurs avec leur rôle et de pouvoir les rechercher. Seulement les admins peuvent accéder à cette page, les utilisateurs ont seulement une page disant Bienvenue.

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

La requête fetch, permet de récupérer tous les utilisateurs par rapport à l’email. La fonction searchUsers, permet d’effectuer des recherches selon l’email (trim : pour enlever les espaces au début et fin). S’il n’y a rien dans la barre de recherche, cela affiche tous les users.

## Protection contre les injections SQL :

Il faut se protéger contre les injections SQL, car attaquant pourrait injecter du code malveillant tel que OR 1=1 --, qui permettrait d’accéder à toutes les données de la table.

Une image contenant texte, capture d’écran, Police, ligne

Le contenu généré par l’IA peut être incorrect.

Ce code est sécurisé contre les injections, car le ‘ ?’ empêche l’injection SQL. La valeur email est compris comme données et non comme code.

## Utilisation de bcrypt :

Bcrypt sert à hacher un mot de passe et de le comparer/vérifier avec le mot de passe entré. Durant l’inscription d’un utilisateur, le mot de passe est haché, ce qui permet de protéger le mot de passe, dans un cas ou les données sont dévoilés. Quand un utlisateur se connecte, bcrypt compare le mot de passe hashé et le mot de passe entré par l’utilisateur, si ces mots de passe correspondent, l’utilisateur peut accéder.

# Conclusion

Ce projet m’a permis d’en apprendre plus sur les risques et la sécurisation d’un site, ainsi de l’utilisation d’un token pour l’authentification.

Durant le projet, je trouve que le développement du site à était parfois compliqué, a cause des confusions avec d’autres modules et projets. Cependant, la structure (model, view) est similaire aux autres projets, ce qui permet de mieux s’y habituer.