P-183\_WebStore



(Une image originale représentant le projet)

Table des matières

[1 Spécifications 3](#_Toc308526316)

[1.1 Titre 3](#_Toc308526317)

[1.2 Description 3](#_Toc308526318)

[1.3 Matériel et logiciels à disposition 3](#_Toc308526319)

[1.4 Prérequis 3](#_Toc308526320)

[1.5 Cahier des charges 3](#_Toc308526321)

[1.5.1 Objectifs et portée du projet (objectifs SMART) 3](#_Toc308526322)

[1.5.2 Caractéristiques des utilisateurs et impacts 3](#_Toc308526323)

[1.5.3 Fonctionnalités requises (du point de vue de l’utilisateur) 3](#_Toc308526324)

[1.5.4 Contraintes 3](#_Toc308526325)

[1.5.5 Travail à réaliser par l'apprenti 4](#_Toc308526326)

[1.5.6 Si le temps le permet … 4](#_Toc308526327)

[1.5.7 Méthodes de validation des solutions 4](#_Toc308526328)

[1.6 Les points suivants seront évalués 4](#_Toc308526329)

[1.7 Validation et conditions de réussite 4](#_Toc308526330)

[2 Planification Initiale 4](#_Toc308526331)

[3 Analyse 4](#_Toc308526332)

[3.1 Opportunités 4](#_Toc308526333)

[3.2 Document d’analyse et conception 4](#_Toc308526334)

[3.3 Conception des tests 5](#_Toc308526335)

[3.4 Planification détaillée 5](#_Toc308526336)

[4 Réalisation 5](#_Toc308526337)

[4.1 Dossier de Réalisation 5](#_Toc308526338)

[4.2 Modifications 5](#_Toc308526339)

[5 Tests 5](#_Toc308526340)

[5.1 Dossier des tests 5](#_Toc308526341)

[6 Conclusion 5](#_Toc308526342)

[6.1 Bilan des fonctionnalités demandées 5](#_Toc308526343)

[6.2 Bilan de la planification 5](#_Toc308526344)

[6.3 Bilan personnel 5](#_Toc308526345)

[7 Divers 6](#_Toc308526346)

[7.1 Journal de travail 6](#_Toc308526347)

[7.2 Bibliographie 6](#_Toc308526348)

[7.3 Webographie 6](#_Toc308526349)

[8 Annexes 6](#_Toc308526350)

# Spécifications

## Titre

Webstore

## Description

Création d’un site sécurisé

## Matériel et logiciels à disposition

Un ordinateur standard de la section informatique avec Docker Desktop

## Cahier des charges

### Objectifs et portée du projet (objectifs SMART)

A compléter. Il s’agit d’ébaucher des réponses aux questions de l’acronyme CQQCOQP (Combien, Quoi, Qui, Comment, Où, Quand, Pourquoi)

### Caractéristiques des utilisateurs et impacts

A compléter… Il s’agit d’identifier le(s) profil(s) de(s) utilisateur-trice(s) type, et les conséquences que cela va avoir sur la conception (couleurs, ergonomie, utilisation, etc.)

### Fonctionnalités requises (du point de vue de l’utilisateur)

A compléter par une espèce de mode d’emploi du produit. S’il s’agissait d’une montre, décrire qu’à part l’heure, il y aura la possibilité d’utiliser un chronomètre, un réveil, …

### Contraintes

Sécurité, backups, disponibilité, système utilisé, interfaces avec autres logiciels, etc.

### Travail à réaliser par l'apprenti

Décrire à quoi doit ressembler le travail produit, ce qu’il faudra rendre …

### Si le temps le permet …

Objectifs complémentairesau cas où le projet n’est pas assez ambitieux dans le temps imparti…

### Méthodes de validation des solutions

Comment les tests vont être entrepris, quels tests doivent être entrepris, etc.…

## Les points suivants seront évalués

* Le rapport
* 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

# Réalisation

## Dockerisation des services

Lors de la mise en place de notre projet, plusieurs fichiers nous ont été fourni. Premièrement un fichier docker-compose.yml qui servira à créer trois containers :

* Mysql (Pour la base de données)
* PhpMyAdmin (Pour gérer la base de données avec une interface)
* NodeJs (Pour notre application, il faut tenir compte que le docker-compose appèle le dockerfile de nodejs)

Le container de nodejs est créer grâce à deux étape, tout d’abord le fichier nodejs.dockerfile permet de créer l’image du container. Puis le dockercompose appelle ce fichier pour que celui-ci se lance avec les deux autres.

Pour mettre en place ces différent container nous devons nous rendre dans le dossier où se situe le fichier docker-compose.yml puis dans un terminal de commande exécuter la commande suivante.

*Docker compose up -d*

Ces trois containers vont interagir entre eux, spécialement le container de notre application et le container de base de données, car nous allons établir une connexion pour que notre application puisse interagir avec celle-ci.

Pour ce faire, nous devons installer les package *mysql2* avec *npm*, puis dans notre application établir la connexion.

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

Description générée automatiquement

Tout d’abord nous importons *mysql2* puis nous créons la connexion grâce à *mysql.createConnection()*, en insérant les bonne information de notre container ainsi que notre base de données (Attention la base de données doit être créer sinon la connexion de pourra pas se faire).

L’host, correspond à l’Ip du container de base de données. Nous pouvons l’obtenir en exécutant la commande *docker inspect <nom-du-container-db>.*

Le port est celui par défaut du container mysql.

## HTTPS

Pour pouvoir accéder à notre site de manière sécurisé avec le protocole HTTPS, nous avons besoin de plusieurs choses.

Tout d’abord nous avons besoin d’une clé et d’un certificat :

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

Description générée automatiquement

Pour récupérer ses deux fichiers, j’ai exécuté la commande suivant avec le CLI de OpenSSL.



Celui-ci va ensuite nous demander des informations pour pouvoirs générer ces deux fichiers. Puis lorsque nous avons générer ses deux fichiers nous pouvons les intégrer a notre application.

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

Description générée automatiquement

Puis dans notre code pour mettre en place le certificat SSL, nous devons importer les deux modules suivants.

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

Description générée automatiquement

Qui vont nous permettre de lire les deux fichier *cert* et *key* et créer la connexion avec le HTTPS.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Pour se faire nous créons une constante *option* qui contient le contenu de nous deux fichiers. Puis nous créons le « server » à l’aide de ses options en spécifiant que nous le lançons sur notre *app* et en écoutant le port 8080.

## Authentification par mot de passe

## Profile du client

Actuel : Pas présent

## Administration

Actuel : Route nécessitant l’authentification pour être accéder

Objectif : Avoir un champs « is admin » pour pouvoir vérifier si l’utilisateur connecter un admin ou pas

## Protection contre les injections SQL



# 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

Date, activité (description qui permet de reproduire le cheminement du projet), durée, liens et références sur des documents externes. Lorsqu’une activité de recherches a été entreprise, il convient d’énumérer ce qui a été trouvé, avec les références.

## Bibliographie

Références des livres, revues et publications utilisés durant le projet.

## Webographie

Références des sites Internet consultés durant le projet.

# Annexes

Listing du code source (partiel ou, plus rarement complet)

Guide(s) d’utilisation et/ou guide de l’administrateur

Etat ou « dump » de la configuration des équipements (routeur, switch, robot, etc.).

Extraits de catalogue, documentation de fabricant, etc.