Documentation Meylan's site

Documentation

Date de création : 23.02.2024  
Version 3 du 22.02.2024

Orlando Espin

|  |  |  |
| --- | --- | --- |
|  |  |  |

Table des matières

[1 Introduction 2](#_Toc125302133)

[2 Domaine de la sécurité Erreur ! Signet non défini.](#_Toc125302134)

[3 Documentation Erreur ! Signet non défini.](#_Toc125302135)

[3.1 Structure d’une documentation Erreur ! Signet non défini.](#_Toc125302136)

[3.2 Etude de cas Erreur ! Signet non défini.](#_Toc125302137)

[3.2.1 Miele **Erreur ! Signet non défini.**](#_Toc125302138)

[3.2.2 Singer **Erreur ! Signet non défini.**](#_Toc125302139)

[3.3 Présentation Erreur ! Signet non défini.](#_Toc125302140)

[3.3.1 Informations utiles et pertinentes **Erreur ! Signet non défini.**](#_Toc125302141)

[3.3.2 Moyens possibles **Erreur ! Signet non défini.**](#_Toc125302142)

[3.4 Prestation Erreur ! Signet non défini.](#_Toc125302143)

[3.4.1 Cycle à 5 étapes **Erreur ! Signet non défini.**](#_Toc125302144)

[a) Mise au courant **Erreur ! Signet non défini.**](#_Toc125302145)

[b) Démonstration **Erreur ! Signet non défini.**](#_Toc125302146)

[c) Mise en situation **Erreur ! Signet non défini.**](#_Toc125302147)

[d) Evaluation du participant **Erreur ! Signet non défini.**](#_Toc125302148)

[e) Evaluation de la prestation **Erreur ! Signet non défini.**](#_Toc125302149)

[4 Le projet Erreur ! Signet non défini.](#_Toc125302150)

[4.1 Réalisation Erreur ! Signet non défini.](#_Toc125302151)

[4.1.1 Présentation personnelle **Erreur ! Signet non défini.**](#_Toc125302152)

[4.1.2 Mise au courant **Erreur ! Signet non défini.**](#_Toc125302153)

[4.1.3 Démonstration **Erreur ! Signet non défini.**](#_Toc125302154)

[4.1.4 Mise en situation **Erreur ! Signet non défini.**](#_Toc125302155)

[4.1.5 Evaluation du participant **Erreur ! Signet non défini.**](#_Toc125302156)

[4.2 Bilan post projet Erreur ! Signet non défini.](#_Toc125302157)

[5 Projet des camarades Erreur ! Signet non défini.](#_Toc125302158)

[5.1 Pour chaque projet, identification des plus et moins Erreur ! Signet non défini.](#_Toc125302159)

[5.1.1 Projet du groupe PVN **Erreur ! Signet non défini.**](#_Toc125302160)

[5.1.2 Projet du groupe East Blue **Erreur ! Signet non défini.**](#_Toc125302161)

[6 Conclusion Erreur ! Signet non défini.](#_Toc125302162)

[6.1 Ce que j’ai aimé Erreur ! Signet non défini.](#_Toc125302163)

[6.2 Ce que j’ai moins aimé Erreur ! Signet non défini.](#_Toc125302164)

# Introduction

Lors de ce projet je vais développer le backend ainsi que le frontend de la partie gestion des utilisateurs pour mon client Claude Meylan. Nous travaillons à deux sur ce site web, mais nous avons divisé les tâches avec qu’il soit en rapport avec ce module. Donc tandis que moi je gérerai les utilisateurs ainsi que toutes leurs actions possible, Adrien gérera le CMS et l’affichage des données sur le site.

# Analyse

## Présentation du projet

Le site web que nous développons est une boutique en ligne dans l’achat de montre de luxe basé à la vallée de joux. Mon rôle sera de faire en sorte que les visiteurs de notre site puissent créer un compte ou se connecter si c’est déjà fait, qu’ils aient accès ou non à certaines pages selon le statuts de leur compte. Ils pourront aussi s’inscrire à une newsletter ainsi que s’en désinscrire. Je dois faire attention à rester dans la légalité au niveau de la gestion des données, donc les clients devront pouvoir en tout temps modifier leurs données ainsi que supprimer leur compte. Je dois leur proposer une solution de suppression de compte efficace afin qu’il ne reste pas de trace d’eux. Chaque client pourra aussi commander des montres et accéder à leur historique des commandes.

## Uses Case

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

Description générée automatiquement

## Diagramme activité

### Création de compte

Une image contenant texte, diagramme, ligne, Parallèle

Description générée automatiquement

### Inscription newsletter

Une image contenant texte, capture d’écran, diagramme, Parallèle

Description générée automatiquement

## Diagramme de séquences systèmes

### Création compte

Une image contenant texte, capture d’écran, diagramme, Parallèle

Description générée automatiquement

## Schéma ER



## Maquettes

### Connexion

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

Description générée automatiquement

### Création de compte

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

Description générée automatiquement

### Infos du compte

Une image contenant texte, capture d’écran, conception

Description générée automatiquement

# Conception

## Diagrammes de classe

### Client

Une image contenant texte, diagramme, capture d’écran, Plan

Description générée automatiquement

### Serveur

Une image contenant texte, diagramme, ligne, Plan

Description générée automatiquement

## Schéma relationnel

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

Description générée automatiquement

## Diagramme séquence interactions

Une image contenant texte, diagramme, Parallèle, Plan

Description générée automatiquement

## Conception des tests

|  |  |  |
| --- | --- | --- |
| Test | Résulat Attendu | Résultat obtenu |
| Connexion avec les bons credentials | Connexion réussie |  |
| Connexion avec les mauvais credentials | Connexion échouée |  |
| Créer un compte avec données valide | Création réussie |  |
| Créer un compte avec donnée non-valide | Création échouée |  |
| Créer un compte avec un mail déjà utilisé | Création échouée |  |
| Acheter une montre | Achat réussi |  |
| Modifier des infos du compte avec des données valides | Modification réussie |  |
| Modifier des infos du compte avec des données non-valide | Modification échouée |  |
| Supprimer son compte | Suppression réussie de la db et déconnexion |  |
| S’inscrire à la newsletter en tant qu’invité | Ajout du mail à la table newsletter |  |
| S’inscrire à la newsletter en tant qu’user | Changement à true du champs newsletter dans la table user. |  |
| Se déconnecter de la newsletter en tant qu’user | Changement à false du champs newsletter dans la table user. |  |
| Se déconnecter de le newsletter en tant que visiteur | Suppression du mail dans la newsletter |  |
| Déconnexion | Suppression de la session et déconnexion coté client. |  |
| Injection SQL | Injection patch |  |
| Injection script/html | Injection patch |  |

# Implémentation

## Descente de code

### Récupération des requêtes

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

    switch ($\_POST['action']) {

        case 'addMontre':

            // $imgBackground = $\_POST['imgBackground'];

            // $imgMontreShadow = $\_POST['imgMontreShadow'];

            // $imgMontre = $\_POST['imgMontre'];

            //Attention rajouter les images après ainisi que la collection id

            if (isset($\_POST['titre'], $\_POST['titreLigne2'], $\_POST['txtDescription\_fr'], $\_POST['txtDescription\_en'], $\_POST['taille'], $\_POST['epaisseur'], $\_POST['mouvement'], $\_POST['prix'], $\_POST['variante'])) {

                $titre = $\_POST['titre'];

                $titreLigne2 = $\_POST['titreLigne2'];

                $txtDescription\_fr = $\_POST['txtDescription\_fr'];

                $txtDescription\_en = $\_POST['txtDescription\_en'];

                $taille = $\_POST['taille'];

                $epaisseur = $\_POST['epaisseur'];

                $mouvement = $\_POST['mouvement'];

                $prix = $\_POST['prix'];

                $variante = $\_POST['variante'];

                //Attention ne contrôle pas que les values ne sont pas vide, a implémenter.

                $montre = new Montre($titre, $titreLigne2, $taille, $epaisseur, $mouvement, $prix, $variante);

                $montre->setTxtDescriptionFr($txtDescription\_fr);

                $montre->setTxtDescriptionEn($txtDescription\_en);

                if (isset($\_POST['collectionID'])) {

                    $collectionID = $\_POST['collectionID'];

                    $montre->setCollectionId($collectionID);

                    $ctrlCms->addMontreCollection($montre);

                } else {

                    $ctrlCms->addMontre($montre);

                }

            } else {

                http\_response\_code(400);

                break;

            }

            break;

### Ajout client

#### Main

        case 'addUser':

            if (isset($\_POST['nom'], $\_POST['prenom'], $\_POST['civilite'], $\_POST['mail'], $\_POST['newsletter'], $\_POST['motDePasse'])) {

                $nom = $\_POST['nom'];

                $prenom = $\_POST['prenom'];

                $civilite = $\_POST['civilite'];

                $mail = $\_POST['mail'];

                $newsletter = $\_POST['newsletter'];

                $motDePasse = $\_POST['motDePasse'];

                $user = new User($nom, $prenom, $civilite, $mail, $newsletter, $motDePasse);

                $ctrlUserManager->addUser($user);

                break;

            } else {

                http\_response\_code(400);

                echo 'ici';

            }

            break;

#### Ctrl

    public function addUser($user)

    {

        if (filter\_var($user->getMail(), FILTER\_VALIDATE\_EMAIL)) {

            if (!empty($user)) {

                if ($this->wrkManagerDBUser->addUser($user) == true) {

                    http\_response\_code(200);

                } else {

                    //code http pour accès non-autorisé

                    http\_response\_code(409);

                }

            } else {

            }

        }

    }

#### Wrk

    public function addUser($user)

    {

        $safeMail = htmlspecialchars($user->getMail(), ENT\_QUOTES, 'UTF-8');

        if ($this->connexion->selectSingleQuery('SELECT \* FROM t\_user WHERE mail = ?', [$safeMail]) == null) {

            if (

                $this->connexion->executeQuery('INSERT INTO `t\_user` (`nom`, `prenom`, `civilite`, `mail`, `newsletter`, `motDePasse`)

         VALUES (?,?,?,?,?,?);'

                    ,

                    [$user->getNom(), $user->getPrenom(), $user->getCivilite(), $user->getMail(), $user->getNewsletter(), password\_hash($user->getMotDePasse(), PASSWORD\_DEFAULT)]

                )

            ) {

                return true;

            } else {

                return false;

            }

        } else {

            return false;

        }

    }

### Connexion

#### Frontend template

<template>

    <div class="login-container">

        <back />

        <form class="account-login">

            <h2 class="teaser-title">Connection</h2>

            <div class="form-item-container">

                <label for="email">Email</label>

                <input v-model="email" type="email" name="email" value="">

            </div>

            <div class="form-item-container">

                <label for="password">Password</label>

                <input v-model="password" type="password" name="password" value="">

            </div>

            <p class="error-message">{{ errorMessage }}</p>

            <RouterLink to="/accountCreate" class="absolute bottom-8 text-sm">don't have an account?</RouterLink>

            <blueButtonSmaller @click="login()" buttonText="connection"

                class="connection-button absolute bottom-0 right-0" />

        </form>

    </div>

</template>

<script setup>

import { ref } from 'vue';

import ServiceHttp from "../../services/ServiceHttp.js";

import blueButtonSmaller from '../../components/blueButtonSmaller.vue';

import back from '../../components/buttonBack.vue';

import router from '../../router/index.js'

</script>

<script>

export default {

    components: {

    },

    data() {

        return {

            email: '',

            password: '',

            errorMessage: '',

            serviceHttp: new ServiceHttp()

        }

    },

    methods: {

        async login() {

            try {

                if (this.email.trim() && this.password.trim()) {

                    await this.serviceHttp.checkLogin(this.email, this.password, this.successLogin.bind(this), this.errorLogin.bind(this));

                } else {

                    this.errorMessage = 'Veuillez remplir tout les champs'

                }

            } catch (Exception) {

                console.log(Exception);

                this.errorMessage = 'Erreur de connexion avec nos serveurs. Veuillez réessayer plus tard.'

            }

        },

        successLogin() {

            this.$router.push("/AccountInfos");

        },

        errorLogin(error) {

            console.log("Je suis une erreur")

            switch (error) {

                case 401:

                    this.errorMessage = 'Vos identifiants sont faux'

                    break;

                case 400:

                    this.errorMessage = "Votre adresse mail n'est pas valide"

                    break;

                default:

                    this.errorMessage = "Erreur de connexion avec nos serveurs. Veuillez réessayer plus tard."

            }

        }

    }

}

#### Frontend httpservice

  deconnectionUser(successCallback, errorCallback) {

    $.ajax({

      type: "POST",

      url: this.URL,

      data: "action=deconnectionUser",

      xhrFields: {

        withCredentials: true,

      },

      success: successCallback,

      error: errorCallback,

    });

  }

#### Backend Main

        case 'connectionUser':

            if (isset($\_POST['mail'], $\_POST['motDePasse'])) {

                $mail = $\_POST['mail'];

                $motDePasse = $\_POST['motDePasse'];

                $ctrlUserManager->connection($mail, $motDePasse);

            }

#### Backend Ctrl

    public function connection($mail, $mdp)

    {

        $user = $this->wrkManagerDBUser->getUserByMail($mail);

        if ($user != false) {

            if (filter\_var($mail, FILTER\_VALIDATE\_EMAIL)) {

                if (password\_verify($mdp, $user->getMotDePasse())) {

                    $this->sessionManager->set("user", $user);

                    http\_response\_code(200);

                } else {

                    //code http pour accès non-autorisé

                    http\_response\_code(401);

                }

            } else {

                //code http pour une requête fausse

                http\_response\_code(400);

            }

        } else {

            //code http pour accès non-autorisé

            http\_response\_code(401);

        }

    }

#### Backend Wrk

    public function getUserByMail($mail)

    {

        // Échappe les caractères spéciaux pour se protéger contre les XSS

        $safeMail = htmlspecialchars($mail, ENT\_QUOTES, 'UTF-8');

        $userArray = $this->connexion->selectSingleQuery('SELECT \* FROM `t\_user` WHERE `mail` = ?', [$safeMail]);

        if ($userArray == null) {

        } else {

            extract($userArray);

            $user = new User($nom, $prenom, $civilite, $mail, $newsletter, $motDePasse);

            if ($user != null) {

                $user->setId($pk\_user);

                array\_walk\_recursive($user, function (&$item) {

                    $item = htmlspecialchars($item, ENT\_QUOTES, 'UTF-8');

                });

                return $user;

            }

        }

    }

## Problèmes rencontrés

### Session réinitialisé entre chaques requêtes

Nous avons rencontré un problème qu’est que nous n’arrivons pas à stocker un user dans la session. Une fois stocké, lorsque nous refaisons une requête censer aller chercher l’user dans la session il est introuvable. Nous avons différentes hypothèses :

* Plusieurs instance de la class session ont été créé.
* La session se recrée automatiquement entre chaques requêtes ce qui supprime ce qui se trouve dans le cache
* Une ligne de code reset à en continue la key user.

### Distri wsl déconnecté de docker

Lorsque je suis allé coder aujourd’hui, mon frontend ne trouvais plus mon backend. Sachant qu’il s’agissait d’une erreur déjà rencontré précédemment, je suis aller recréer l’image de mon backend et refaire un container. Mais là au moment de créer le container, wsl ne trouvait pas linux. Il m’a fallu une bonne demi-heure de recherche pour comprendre que dû à je ne sais quelles raison, le paramètre de connexion à wsl s’était décoché dans docker. Une fois ce paramètre recocher le problème a été réglé.

## Tests fonctionnels

Les test fonctionnels ci-dessous ont été fait avec Postman dans l’attente de pouvoir faire fonctionner mon frontend chez moi.

|  |  |  |
| --- | --- | --- |
| Test | Résulat Attendu | Résultat obtenu |
| Connexion avec les bons credentials | Connexion réussie | Connexion réussie |
| Connexion avec les mauvais credentials | Connexion échouée | Connexion échouée |
| Créer un compte avec données valide | Création réussie | Création réussie |
| Créer un compte avec donnée non-valide | Création échouée | Création échouée |
| Créer un compte avec un mail déjà utilisé | Création échouée | Création échouée |
| Acheter une montre | Achat réussi | Pas encore implémenter api stripe coté frontend |
| Modifier des infos du compte avec des données valides | Modification réussie | Modification réussie |
| Modifier des infos du compte avec des données non-valide | Modification échouée | Modification échouée |
| Supprimer son compte | Suppression réussie de la db et déconnexion | Suppression réussie de la db et déconnexion |
| S’inscrire à la newsletter en tant qu’invité | Ajout du mail à la table newsletter | Finalement la table newsletter à été supprimer car elle était intégrer de base au site que j’utilise pour gérer la newsletter avec leur sdk. |
| S’inscrire à la newsletter en tant qu’user | Changement à true du champs newsletter dans la table user. | Changement à true du champs newsletter dans la table user. |
| Se déconnecter de la newsletter en tant qu’user | Changement à false du champs newsletter dans la table user. | Changement à false du champs newsletter dans la table user. |
| Se déconnecter de le newsletter en tant que visiteur | Suppression du mail dans la newsletter | Fonction abandonner car trop compliquée à mettre en place et possible à faire depuis le mail directement |
| Déconnexion | Suppression de la session et déconnexion coté client. | Nous avons géré nos sessions différemment elle ne se supprimer donc pas mais le use case est fonctionnel. |
| Injection SQL | Injection patch | Injection patch |
| Injection script/html | Injection patch | Injection patch |

## Hébergement

# Synthèse

## Test à faire pour monsieur Meylan

Alors pour tester que mes fonctionnalité marchent je vais vous faire une marche à suivre :

1. Premièrement accéder au site web via l’url :
2. Deuxièmement rajouter à l’url « /loginCms ». Cela va vous emmener à la page admin.
3. Vous allez entrer les credentials : admin et Pa$$w0rd. Cela va permettre de démontrer le fonctionnement la requête post de connexion.
4. Vous aurez ensuite accès au CMS, lorsque vous naviguerez dedans, vous ne le verrez pas mais entre chaque changement de page, une requête get est envoyé au serveur pour vérifier que vous être un user autoriser à être dans ces pages.
5. Finalement vous pourrez retourner à la page principal du site et vous verrez une requête qui vous déconnecte du user cms.

## Présentation réalisation

La réalisation a été assez compliquer dû aux nombres de choses à faire qui ne rentrait pas dans le cadre du module.

En effet ce site, étant commandé par un client, à du implémenter pleins d’autres fonctionnalités qui ont pris plus ou moins de temps. Cependant la réalisation du projet à été très intéressante. J’ai pu acquérir une grosse expérience à force de gérer des erreurs matin midi et soir pendant trois mois.

La réalisation donc pris plus de temps de prévu, je pensais comprendre plus facilement certaines erreurs et technologies. Je me suis donc un peu surestimé pour ce projet bien qu’il soit mené à bien.

# Conclusion

Le projet m’a amener de nombreuse nouvelles compétence surtout au niveau des technologies, en commençant par php language auquel je n’avais jamais touché avant de commencer ce module.