Documentation Technique

-Thibault Cart-

Table des matières

[**Enoncé**](#_xuaj73nzq46f) **3**

[1 Titre](#_f2if9wdvxwre) 3

[2 Matériel et logiciels à disposition](#_tijalt2b1sqe) 3

[3 Introduction](#_a29bwuho6pr) 3

[4 Descriptif complet du projet](#_ss1dg3oi1tfz) 4

[5 Livrables](#_m9nln6t1fu92) 6

[6 Points techniques évalués spécifiques au projet (obligatoire)](#_l92z3e8jhgs3) 7

[**Analyse fonctionnelle**](#_ax224k78gtfd) **8**

[Présentation du projet:](#_aodwr0dnzysv) 8

[Outils et langages utilisés :](#_9eg2zrexu0z3) 8

[Fonctionnalités :](#_4x4eot8naz1z) 9

[Maquettes:](#_5nasw8ts9mli) 10

[**Analyse organique**](#_1o2hl7ol5u5i) **21**

[Architecture du site](#_ku21511tx5xm) 21

[Sécurité:](#_ytwbkcjf9tuv) 22

[Base de données:](#_u20io829pxh1) 23

[**Protocole et rapport de test**](#_1o2hl7ol5u5i) **23**

[Conclusion rapport de test](#_filbich19evs) 32

[**Conclusion**](#_1o2hl7ol5u5i) **33**

[**Annexes**](#_q4xaw1bht3op) **34**

[Mode d’emploi](#_8hn0x125i4kr) 34

[Code Source](#_8hn0x125i4kr) 34

[Journal de bord](#_8hn0x125i4kr) 34

[Planning initial et final](#_cao25rmn7u7s) 34

[Bibliographie et webographie](#_tcj3hbs3vqx5) 35

[Glossaire](#_cao25rmn7u7s) 35

# Enoncé

## 1 Titre

Outil de collaboration pour le collège d’expert, module Administration

## 2 Matériel et logiciels à disposition

• Un PC standard école, 2 écrans

• Windows 10

• EasyPHP, WSL, autre WAMP ou container Proxmox (PHP 7.x, MySQL ou MariaDB)

• NetBeans ou autre éditeur équivalent

• Suite Office

## 3 Introduction

3.1 Contexte général Le but principal de cette application est de donner aux membres du collège d’experts en informatique du canton de Genève un outil leur permettant de gérer leur travail tout au long de l’année scolaire. Cette application est accessible via le web et est modulaire.

Les modules prévus initialement sont

• L’administration des utilisateurs,

• La rédaction des énoncés des TPIs,

• La répartition des TPIs entre les experts,

• La validation des énoncés des TPIs,

• L’évaluation des TPIs

Le design de l’application est effectué par un groupe d’experts volontaires. Il est en charge de définir la base de données, les modèles de données, la technologie employée ainsi que de développer les CSS de chaque module afin de garantir une cohérence du produit fini.

Le schéma ci-dessous représente l’ensemble des tables du projet. Chaque module n’utilise qu’un sous-ensemble de ces tables.

Le développement est assuré par des apprentis du CFPT informatique lors de leur TPI de fin d’année. Les experts eux- mêmes définissent les cahiers des charges de ces TPIs, en collaboration avec le maître d’apprentissage qui les suit. Il y a globalement un TPI par module.

3.2 Technologie L’application est développée en MySQL - PHP. JavaScript et HTML5 doivent être utilisés pour la validation des formulaires avant soumission.

3.3 Outils Le choix des outils est à la charge de l’apprenti pour autant qu’il puisse développer dans les langages et technologies choisis par les experts.

3.4 Planification Planning approximatif :

• Analyse : 12h

• Implémentation : 36h

• Tests : 16h

• Documentation : 24h

Vous devez établir un planning détaillé avant la fin de la première journée

3.5 Méthodologie Dans le cadre de ce travail, vous devez appliquer une méthodologie pour la gestion de votre projet. A vous de choisir la méthodologie appropriée au cadre du TPI.

## 4 Descriptif complet du projet

4.1 Description fonctionnelle L’application est organisée en arbre dans lequel chaque branche est un module.

La page d’accueil présente l’application, son but et ses fonctionnalités principales. Elle intègre également la liste des modules accessibles, le lien pour y accéder ainsi qu’une description de celui-ci. Les liens ne sont actifs que si la personne est identifiée.

Selon les rôles et fonctions de la personne identifiée, certains modules peuvent apparaître ou pas.

Le menu principal (horizontal) donne accès aux différents modules, le menu secondaire (vertical) donne accès aux fonctionnalités du module sélectionné.

4.2 Module Administration Ce module intègre toutes les données de bases nécessaires à l’administration du site et des données communes à l’application.

• Gestions des paramètres globaux de l’application

• Gestion des utilisateurs et des rôles

• Gestion des classes

• Gestion des droits d’accès aux modules

Afin de centraliser les informations, les utilisateurs sont tous dans une seule table. Chaque utilisateur se voit attribuer un ou plusieurs rôles.

Les rôles sont de type :

• Candidat

• Prof / chef de projet

• Expert

• Administrateur

Un candidat est associé à une classe et une année. En cas de redoublement, il faut le réinscrire pour une nouvelle année.

Tous les autres rôles sont intemporels et se voient attribuer des fonctions spécifiques dans les différents modules.

Dans le module administration, seul le rôle d’admin gère toutes les actions CRUD sur les utilisateurs, les rôles, les apprentis, les classes et les droits d’accès aux modules.

La relation Prof - Apprenti se fait dans le module Rédaction des TPIs et sera réalisé par un autre candidat

4.2.1 En tant qu’administrateur, je dois pouvoir :

• Gérer les paramètres globaux de l’application : année courante, dates de début et de fin de l’année en cours, ainsi que d’autres paramètres qui pourront être utiles aux autres modules (par exemple dates de début et de fin de répartition des TPI)

• Gérer les classes : afficher la liste des classes par année, créer une nouvelle classe (la valeur de l’année par défaut est l’année en cours), modifier le nom de la classe. La suppression d’une classe n’est possible que si aucun candidat n’y a été rattaché.

• Gérer les rôles : cette table est à priori statique. Dans le cadre de ce TPI, seule la lecture des rôles est prévue.

• Gérer les modules, les droits et la liaison avec les rôles : ces tables permettent de tenir compte des modules installés dans l’application. Les données à saisir sont spécifiées dans les documentations techniques, partie installation, des modules concernés.

• Gérer les utilisateurs et les rôles qu’ils peuvent avoir : créer un nouvel utilisateur, remplir l’ensemble des champs, attribuer un mot de passe initial, modifier un utilisateur, attribuer/retirer un ou plusieurs rôles. La suppression d’un utilisateur n’est possible que s’il n’est pas lié à un TPI et qu’il ne s’agit pas de l’utilisateur actuellement connecté.

• Me déconnecter.

4.2.2 En tant qu’autre utilisateur identifié, je dois pouvoir :

• Consulter mes données personnelles, mes rôles, mes classes (pour le candidat). Je dois pouvoir modifier mon mot de passe. Le nouveau sel est généré automatiquement à chaque changement du mot de passe.

• Me déconnecter

4.2.3 En tant qu’utilisateur non identifié, je dois pouvoir :

• M’identifier.

4.3 Gestion de l’utilisateur connecté

• L’utilisateur s’identifie à l’aide de son email et de son mot de passe. Dans le jeu de test fourni, tous les mots de passe sont construits avec les initiales du prénom et du nom suivis des chiffres 1234. (par exemple : Pascal Comminot : pascal.comminot@prof.cfpt.ch / pc1234)

• Toutes les actions applicables dans le cadre de ce projet sont gérées / filtrées à l’aide des droits décrits dans les tables modules, rights, roles, role\_rights et user\_roles. La vue user\_rights fournit la liste des droits pour chaque utilisateur.

• A vous de définir pour votre module Administration quels sont les droits que vous avez besoin pour gérer l’accès aux fonctionnalités, et de les appliquer / vérifier.

## 5 Livrables

• Planning prévisionnel fourni le premier jour du TPI

• Rapport de projet

• Manuel utilisateur

• Journal de travail

• Résumé du travail

## 6 Points techniques évalués spécifiques au projet (obligatoire)

correspondants aux points A14 à A20 du formulaire d’évaluation

A14 : Les fonctionnalités concernant la table users sont implémentées.

A15 : Les fonctionnalités concernant les tables classes et user\_classes sont implémentées

A16 : Les fonctionnalités concernant les tables modules et rights sont implémentées

A17 : Les fonctionnalités concernant les tables user\_roles, roles et role\_rights sont implémentées

A18 : Les modalités d’accès aux diverses fonctions du module sont gérées avec des droits décrits dans la vue user\_rights

A19 : L’application est construite de façon à pouvoir intégrer les divers modules.

A20 : L’application est protégée contre les injections SQL et XSS

# 

# Analyse fonctionnelle

## Présentation du projet:

Dans le cadre de mon TPI et en collaboration avec plusieurs de mes camarades, je vais faire un site web qui permettra une gestion des TPIs uniformisée et simplifiée.

Le projet comporte 5 modules initiaux qui sont:

* Administration des utilisateurs
* Rédaction des TPIs
* Répartition des TPIs entre les experts
* Validation des énoncés
* Evaluation des TPIs

Je dois personnellement m’occuper de la partie administration et sécurité du site.

## Outils et langages utilisés :

**XAMPP:** Application permettant de mettre en place un serveur Web local, elle m’est utile pour exécuter les scripts PHP et héberger la base de données localement. Contrairement à son homologue easyPHP, XAMPP est entièrement gratuit et il est facile d’y installer XDebug

**PhpStorm:** IDE conçu pour permettre l’édition de code php, html, JavaScript. Cette IDE est simple d’utilisation et propose de bonnes fonctionnalitées qui permettent de simplifier l’écriture de code comme le soulignement d’erreur et l’autocomplétion.

**Github:** Logiciel d’hébergement de projet en ligne, il permet de sauvegarder son projet et de gérer les différentes versions de ce dernier de façon simple et intuitive.

**OneDrive:** Microsoft OneDrive est un espace de stockage Cloud que je vais utiliser en complément de GitHub en cas de défaillance de ce dernier.

**PHP:** langage de programmation qui tourne côté serveur, utilisation de la version 7.3 de php

**SQL:** langage permettant d’interagir avec la base de données par le biais de requêtes.

**Bootstrap:** FrameWork html et css , permet de faire des pages web esthétiques et ceci très simplement.

## Fonctionnalités :

**Utilisateur Non identifié**

* **Se connecter**
* **Page Login**

**Administrateur :**

* **Déconnexion**
* **Page gestion des classes**
* **Page gestion des paramètres du site**
* **Page gestion des utilisateurs**

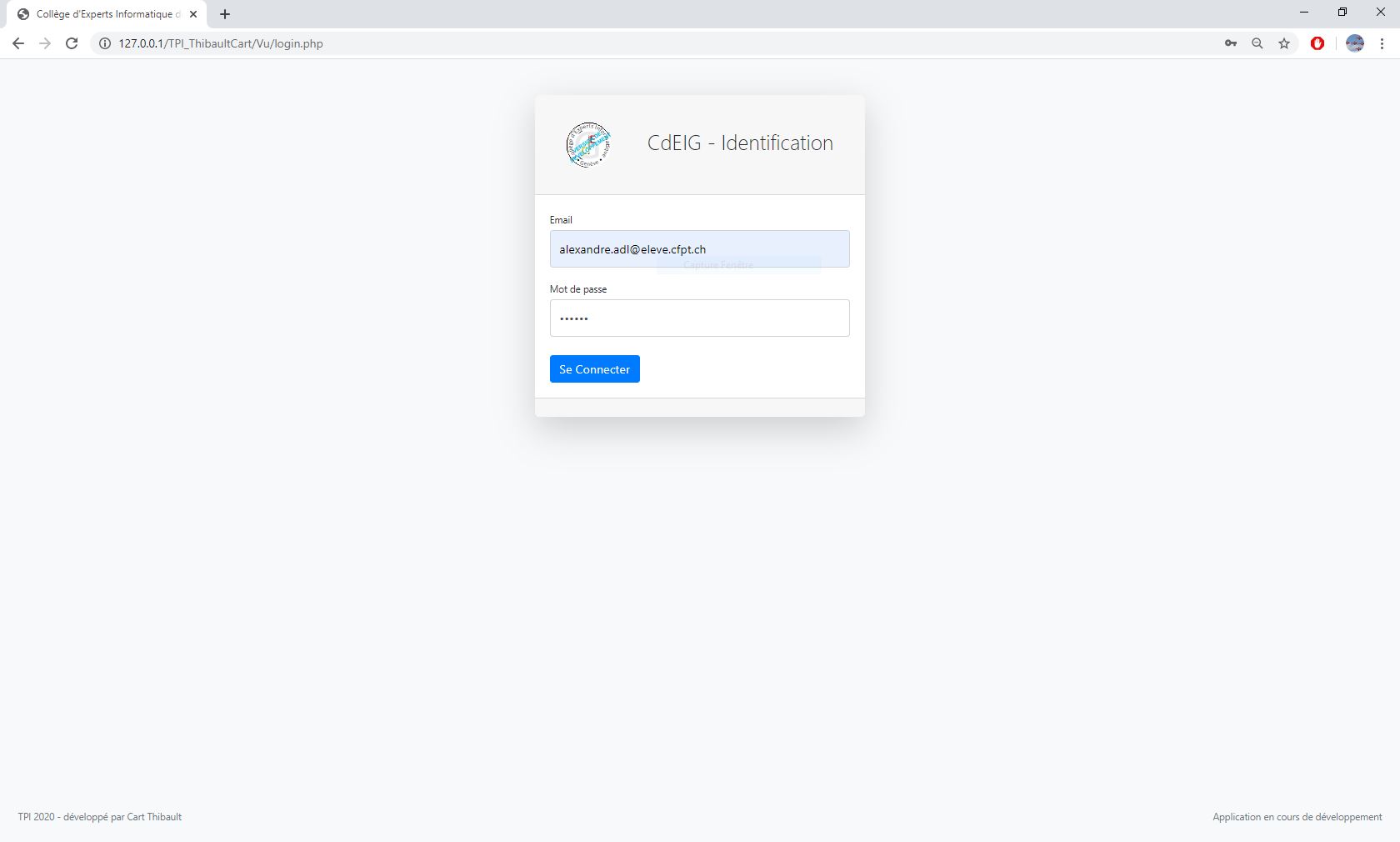
**Utilisateur Connecter**

* **Déconnexion**
* **Page Paramètres**

## 

## Maquettes:

**Page Login**

Page de connexion

Page qui permet de se connecter au site Web via un formulaire. En cas de donnée qui ne correspond à aucun compte, le message suivant s'affiche :”l'adresse mail et le mot de passe ne correspondent à aucun compte”. Je donne volontairement peu d’informations sur l’erreur afin de ne pas faciliter le piratage par force brute .

**Page Home**

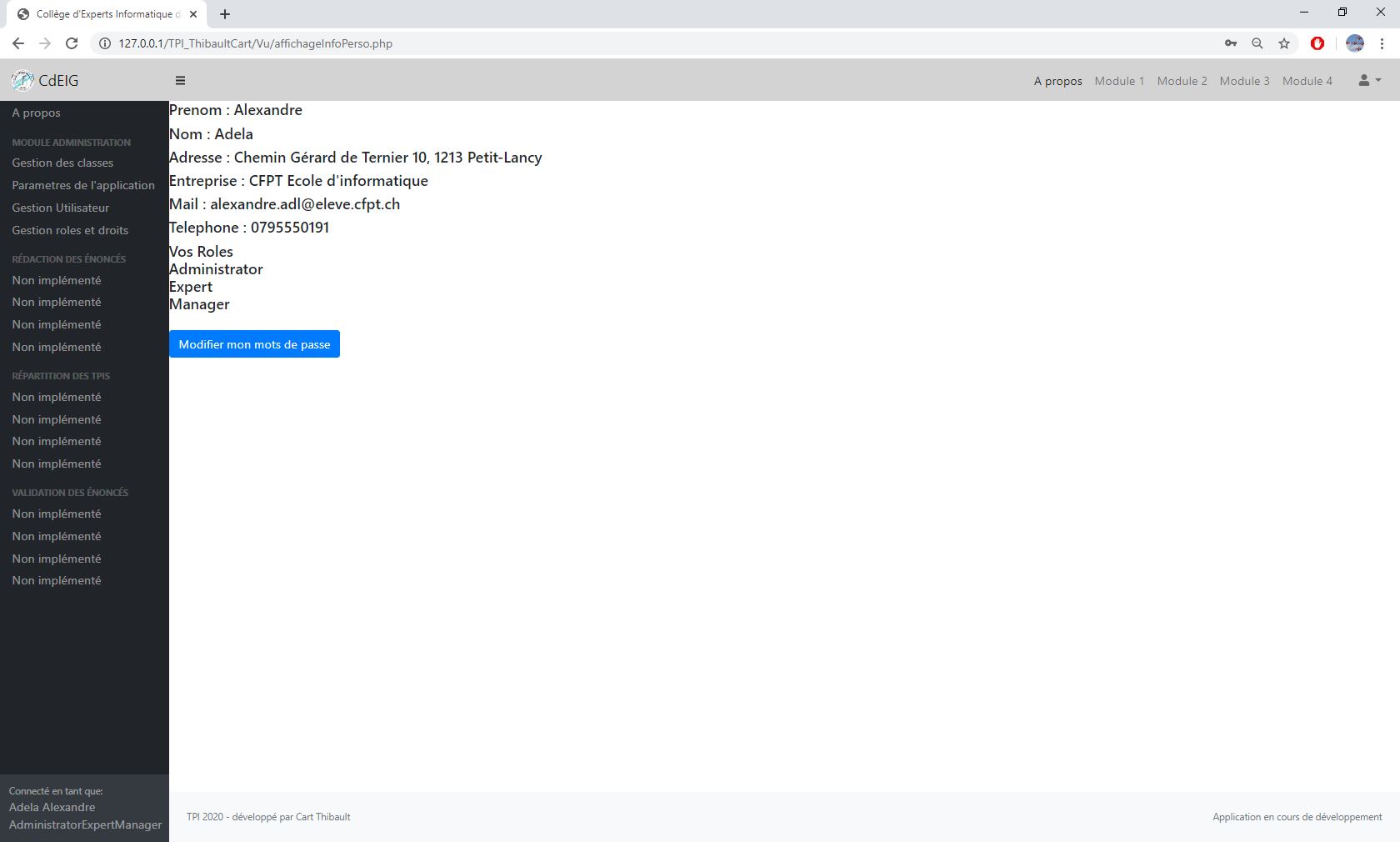
****

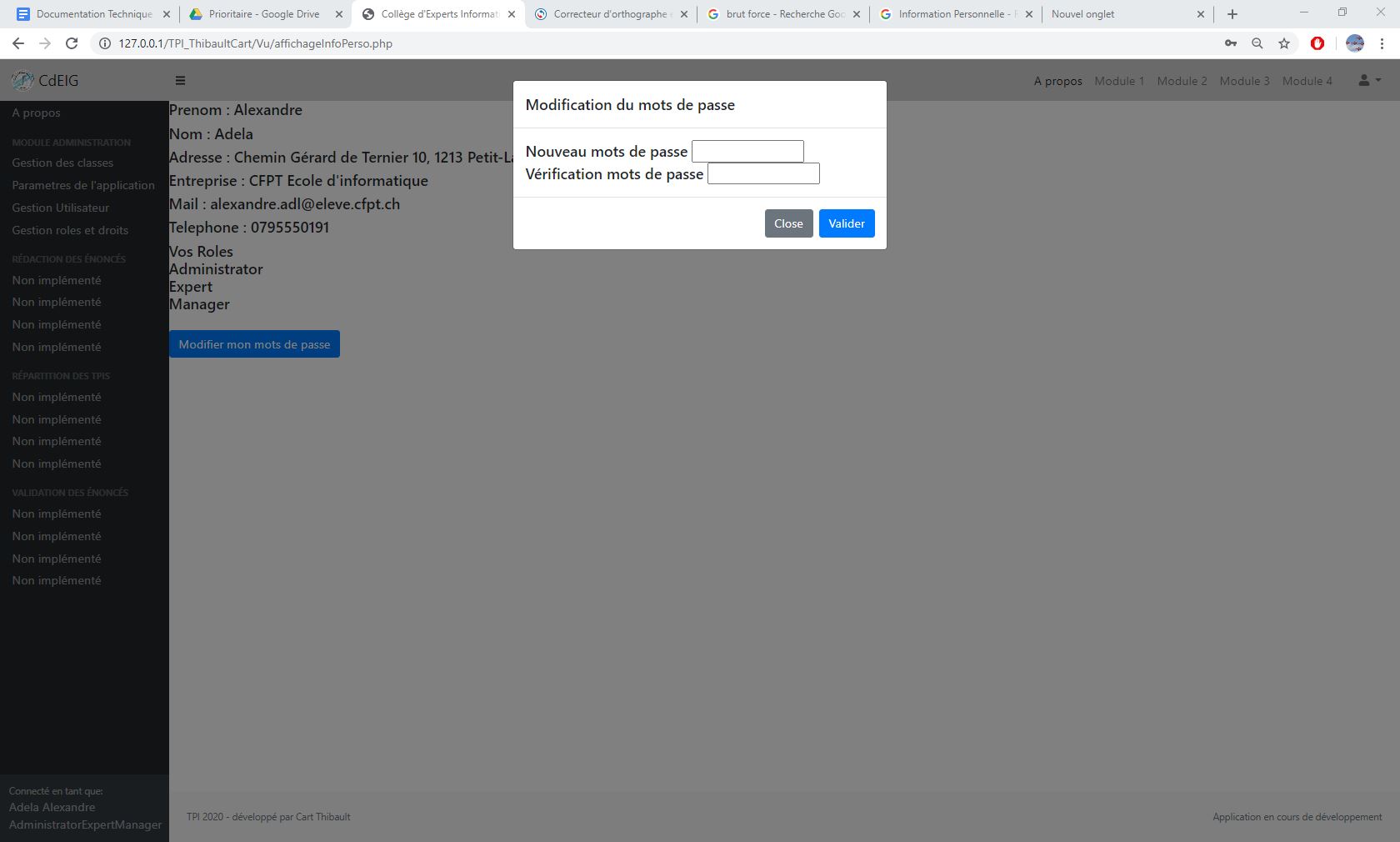
Page Home

Une fois connecté, l’utilisateur est redirigé vers cette page. Une petite explication du site est présente ainsi que la navigation latérale. Nous allons retrouver cette navigation sur toutes les pages du site. En fonction de nos rôles, la navigation va afficher plus ou moins de modules.

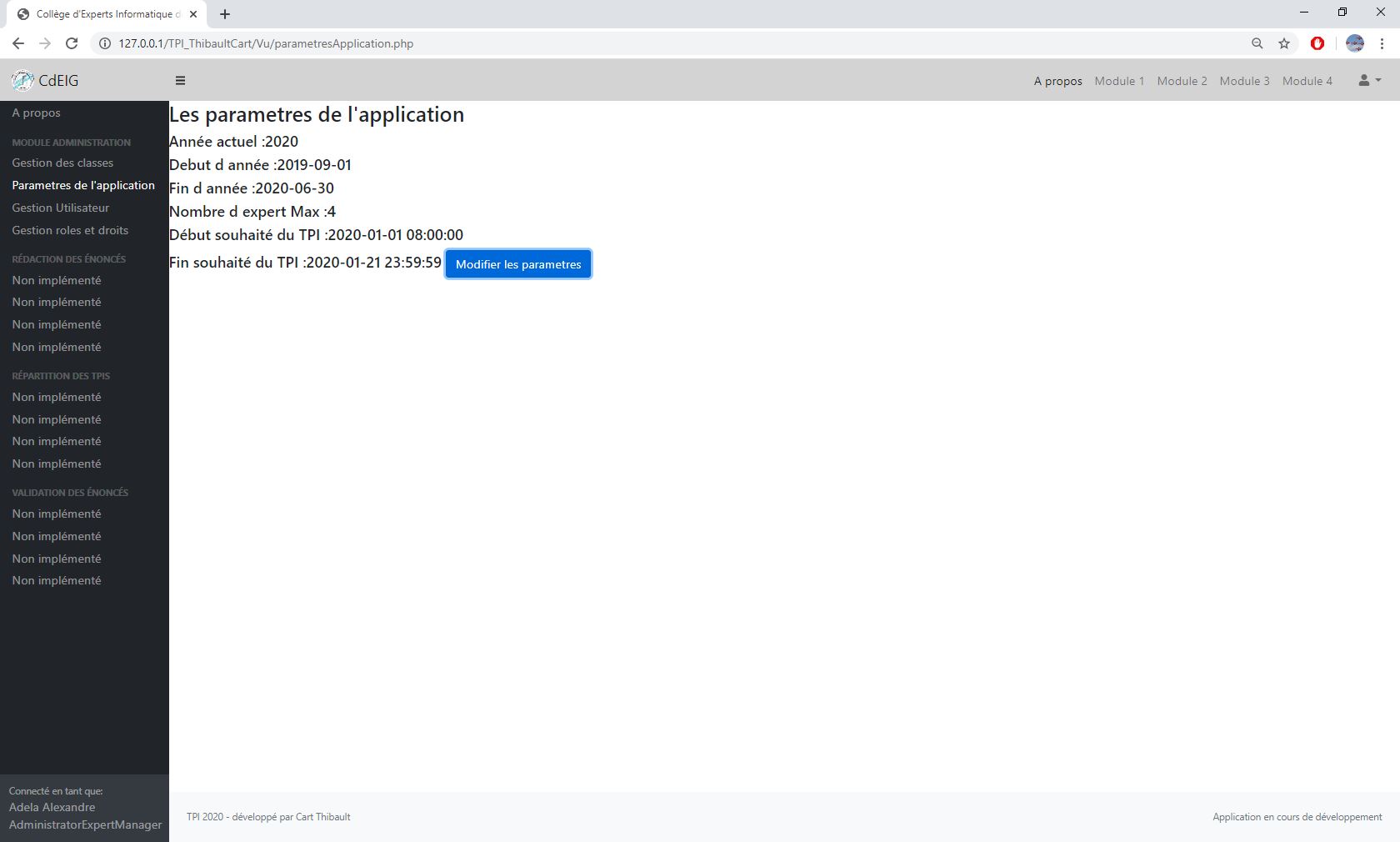
Sur la navigation horizontale nous pouvons accéder à nos informations personnelles et nous déconnecter.

**Page Information Personnelle**

Page Affichage information personnelle

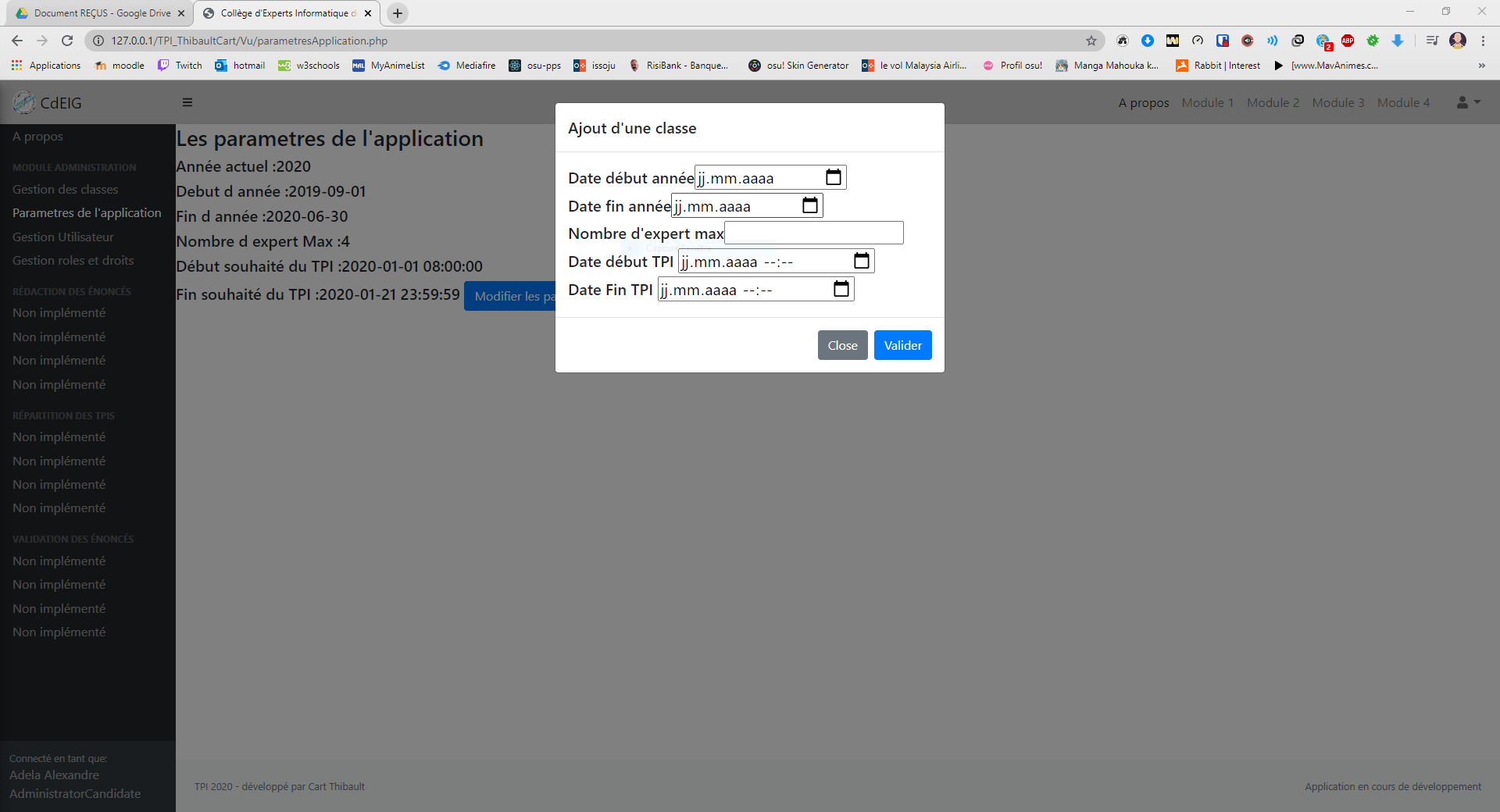
La page affiche nos informations personnelles et permet, grâce à l’appui sur un bouton, de changer notre mot de passe depuis un formulaire modal. Les champs sont protégés contre l’injection de code.

Formulaire modal**Page Paramètres Application**

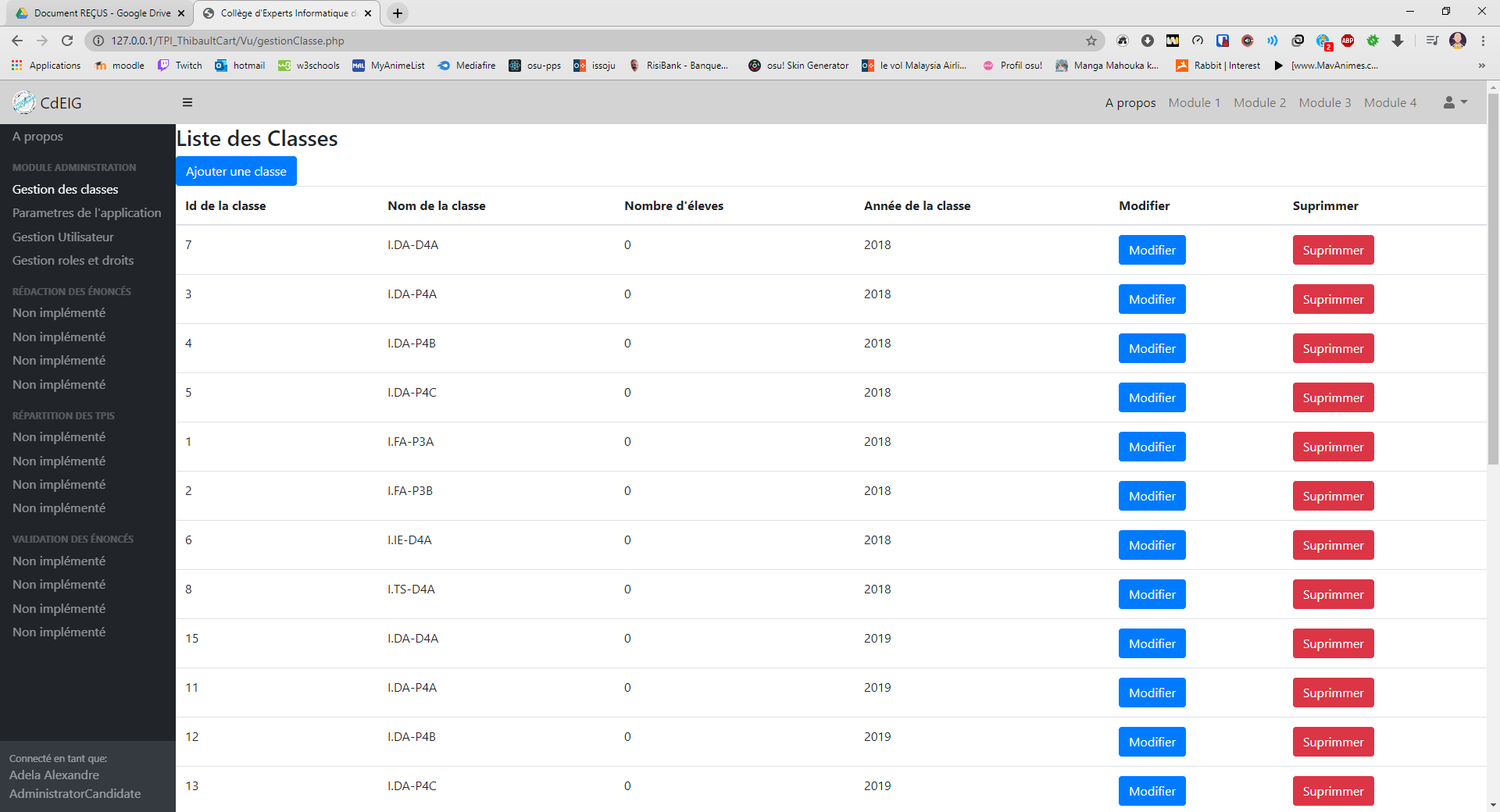


Page Paramètres application

La page montre les paramètres actuels du site, en rapport avec le TPI ou avec l’année en cours. L’appui sur le bouton de modification ouvre un formulaire modal permettant de modifier les paramètres du site.



**Page Gestion des Classes**



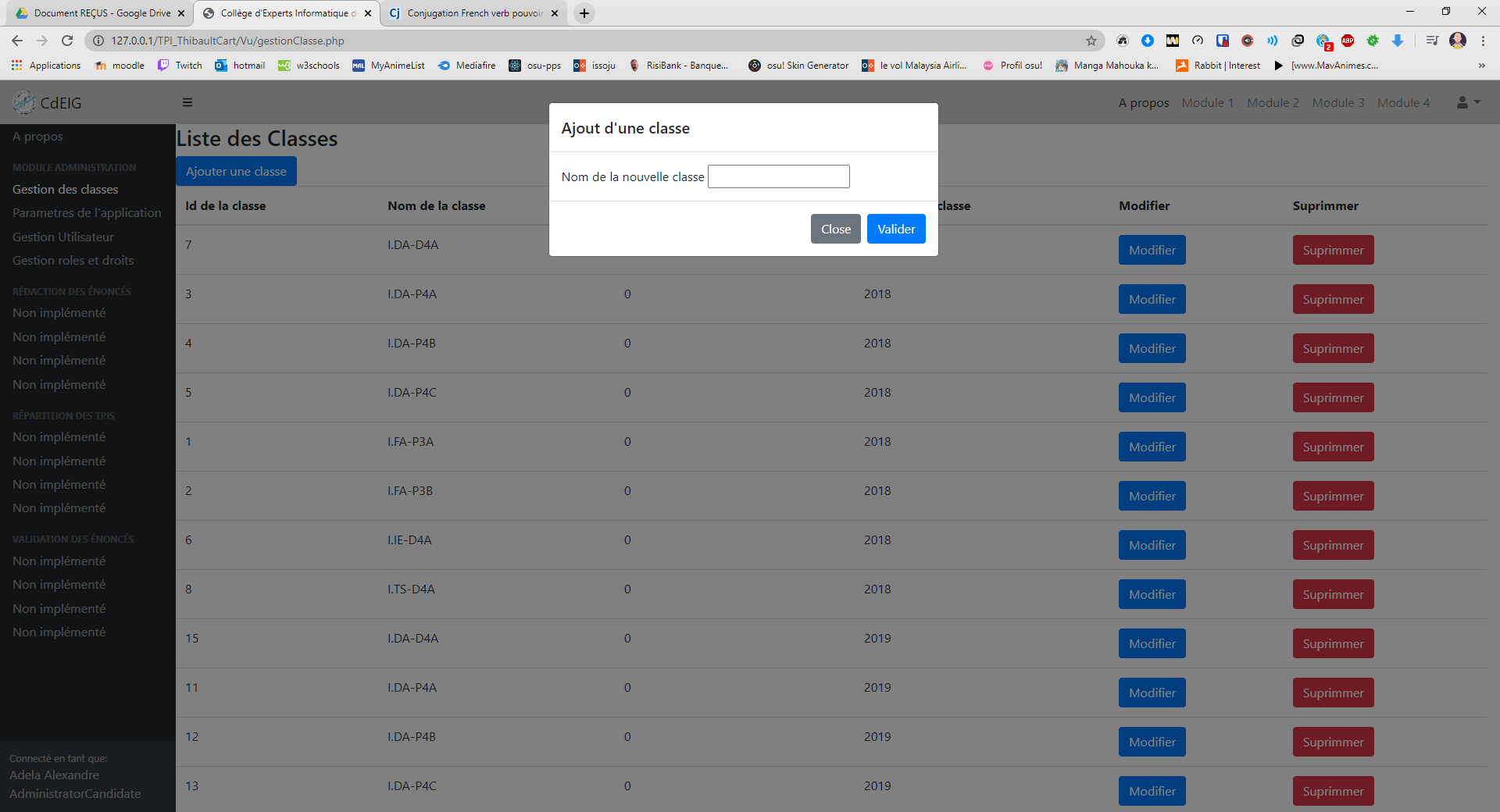
Page Gestion des classes

Page qui affiche toutes les classes, et le nombre d'élève qu’elles contiennent.

Nous avons aussi un formulaire modal qui permet d’ajouter une classe, le formulaire s'ouvre lors de l’appui sur le bouton “Ajouter une classe”.

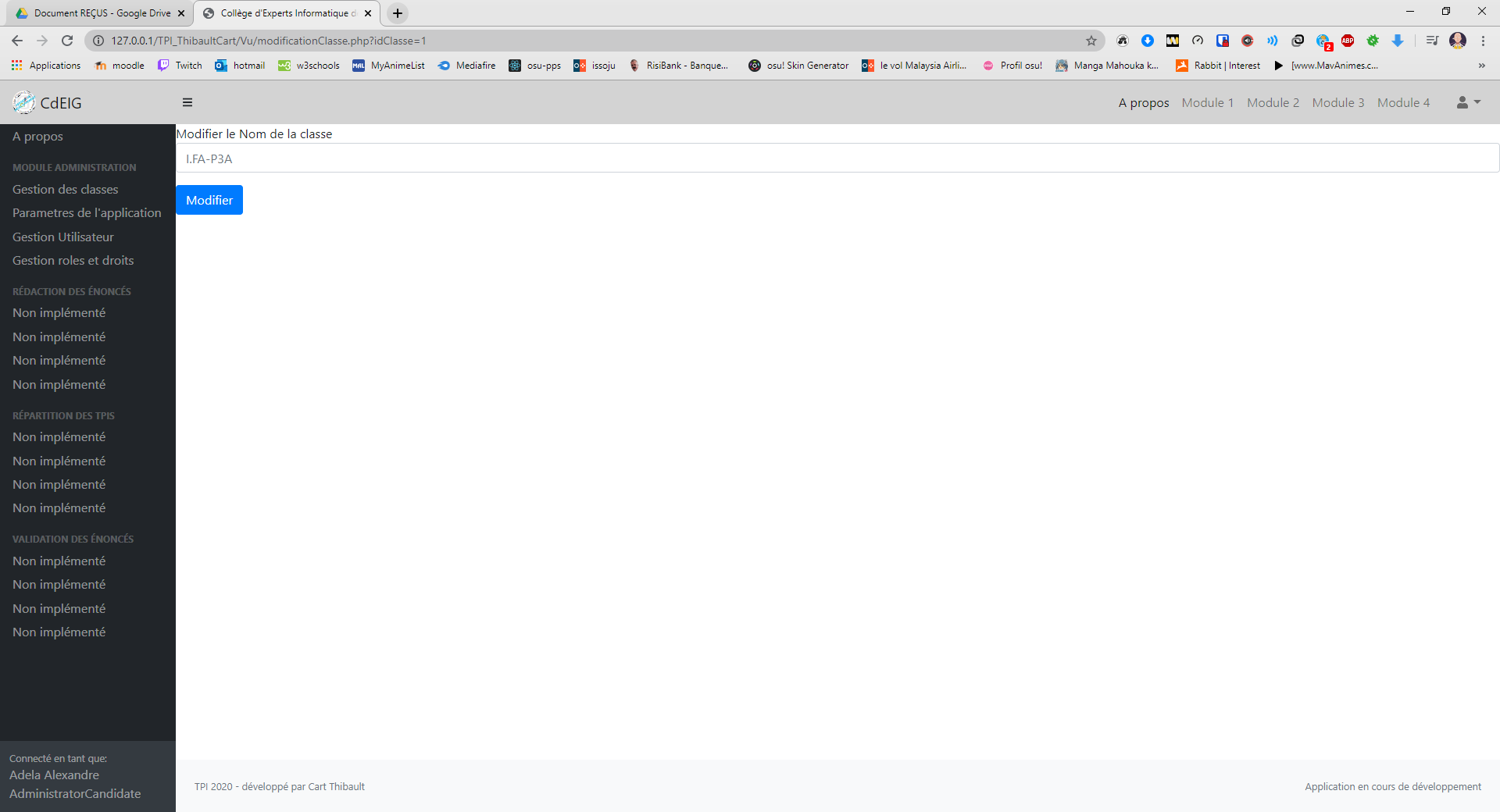
L’appui sur le bouton “Supprimer” supprime la classe à condition qu'elle soit vide.

L’appui sur le bouton “Modifier” redirige vers une autre page où la modification pourra être faite.



Formulaire modal

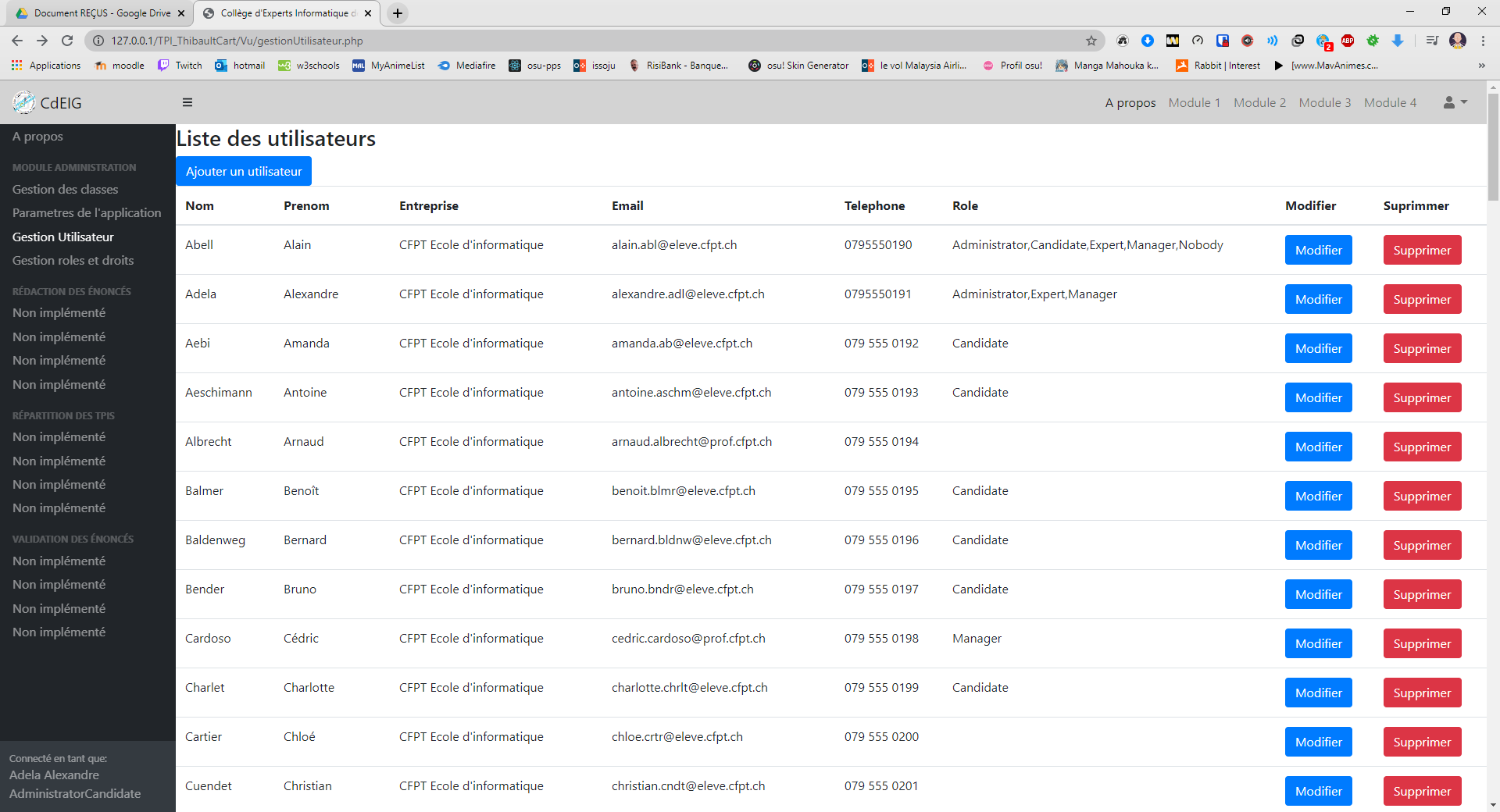
**Page Modification Classe**

****

Page Modification classe

Page qui permet de modifier le nom de la classe, le champ est filtré afin d’éviter l’injection de code. L’ancien nom de la classe est affiché en grisé en arrière plan.

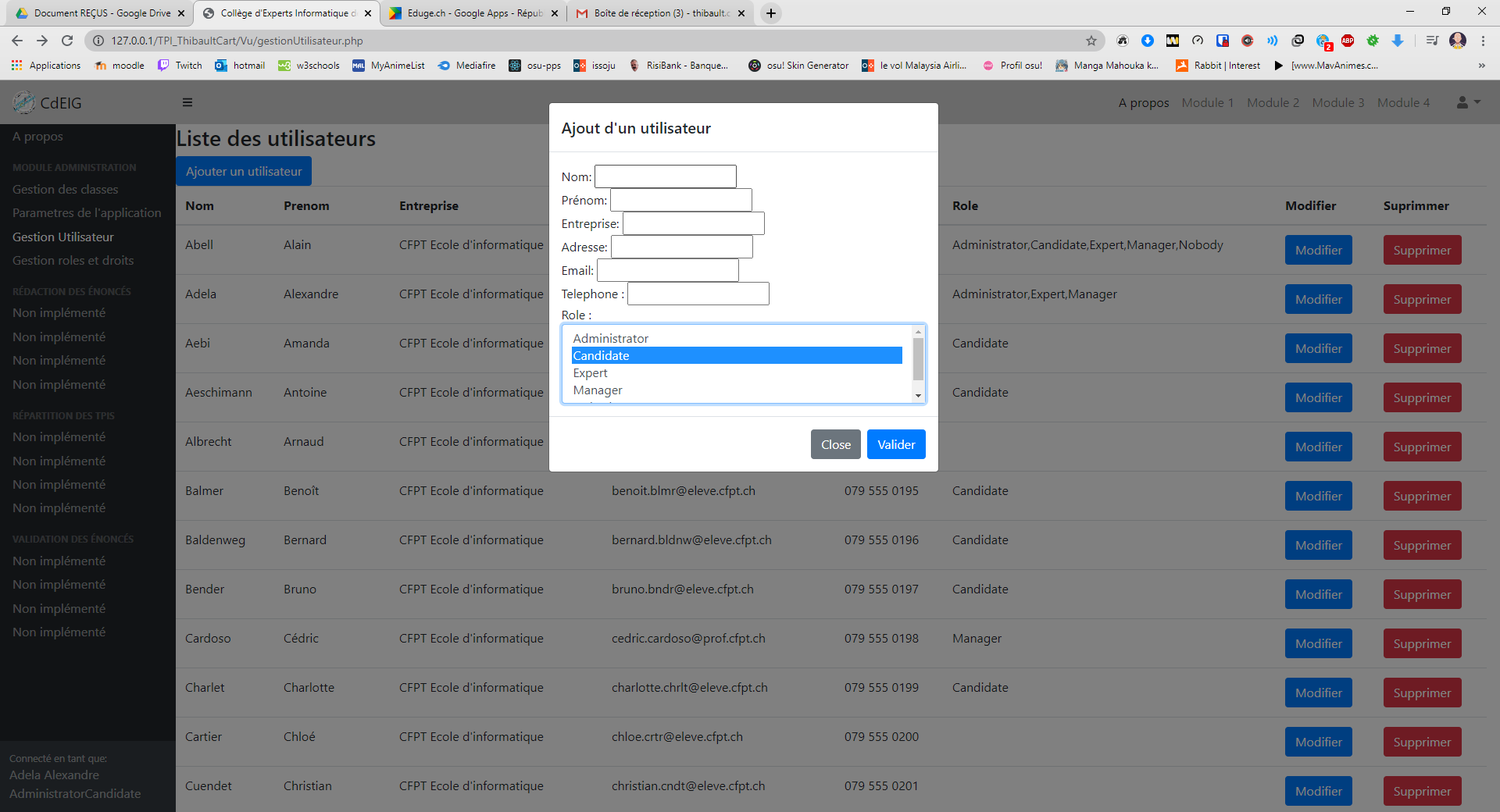
**Page Gestion Utilisateur**

Page Gestion Utilisateur

Page qui affiche les utilisateurs et leurs rôles. Le bouton “Ajouter un utilisateur” ouvre un formulaire modal qui permet d’ajouter un utilisateur.

La bouton “Modifier” ouvre une autre page qui permettra de modifier l’utilisateur.

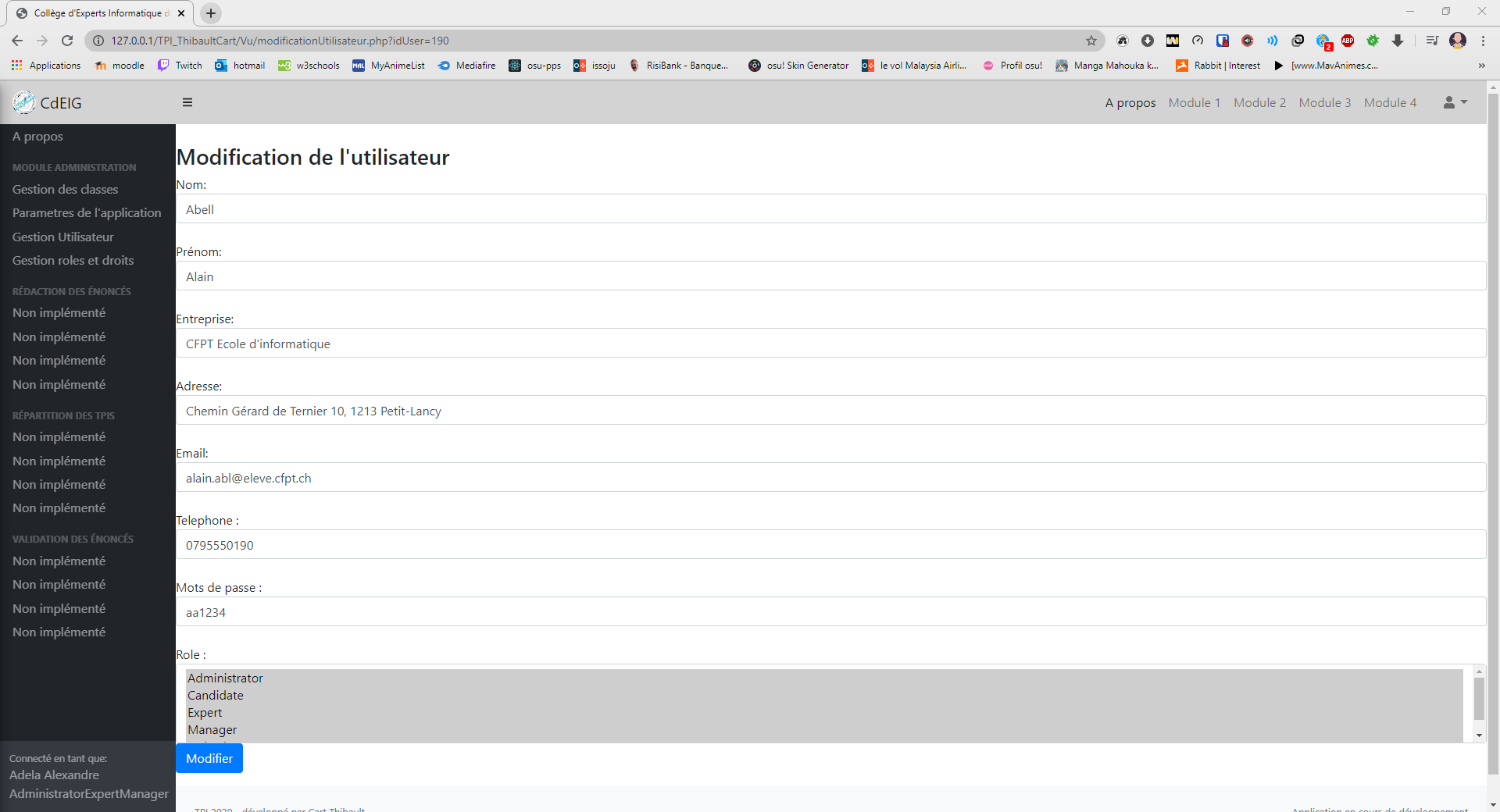
Le bouton “Supprimer” supprime l’utilisateur à condition qu’il ne soit pas relié à un TPI ou qu’il ne soit pas l’utilisateur connecté actuellement.



Formulaire Ajout d’utilisateur

Formulaire modal qui permet d’ajouter un utilisateur, les champs sont filtrés afin d’éviter les injections de code.

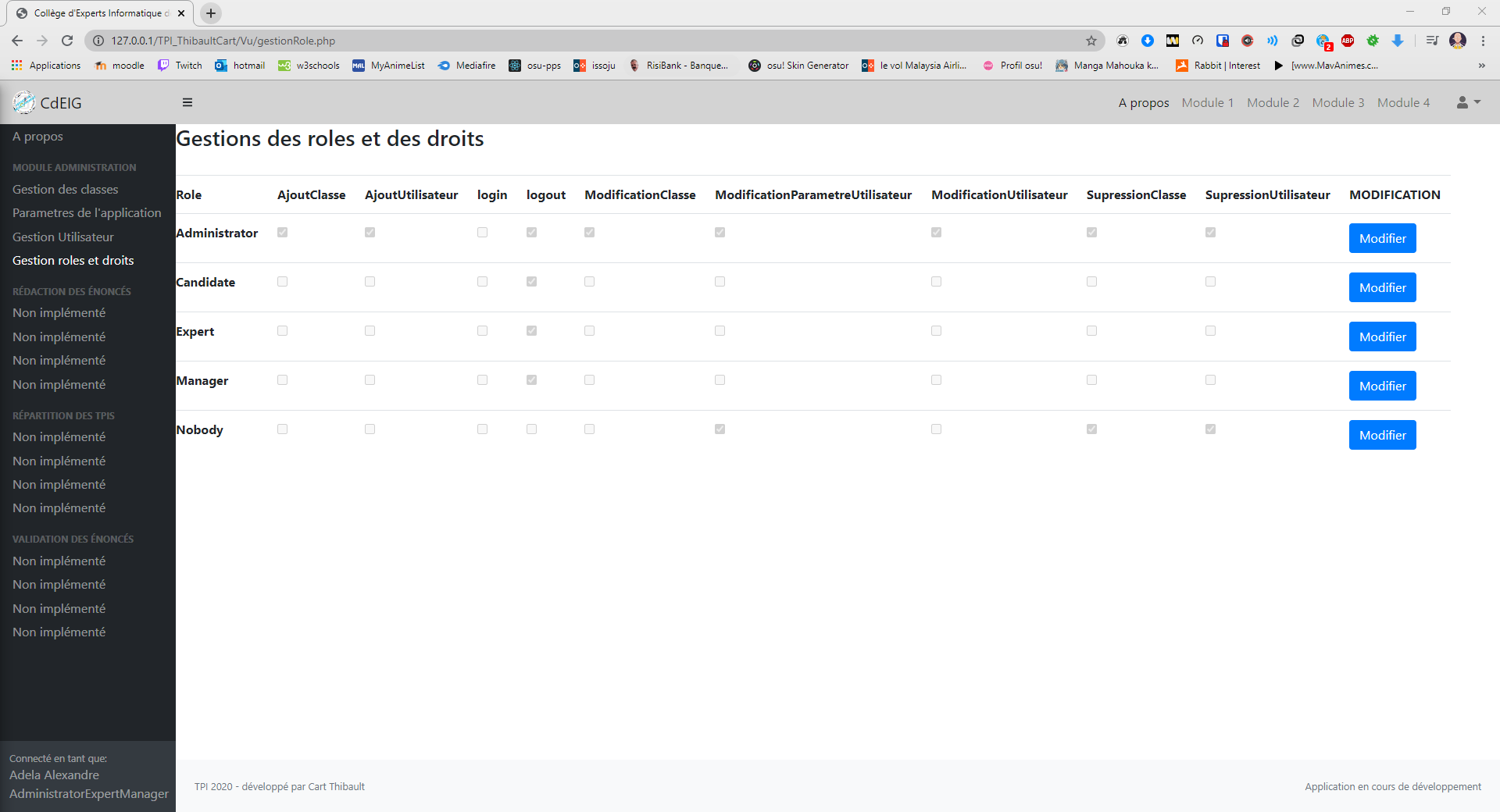
**Page Modification Utilisateur**



Page Modification utilisateur

Cette page permet de modifier les informations de l’utilisateur, les champs sont déjà pré-remplis avec les données de l’utilisateur que nous modifions. Les champs sont filtrés afin d’éviter les injections de code.

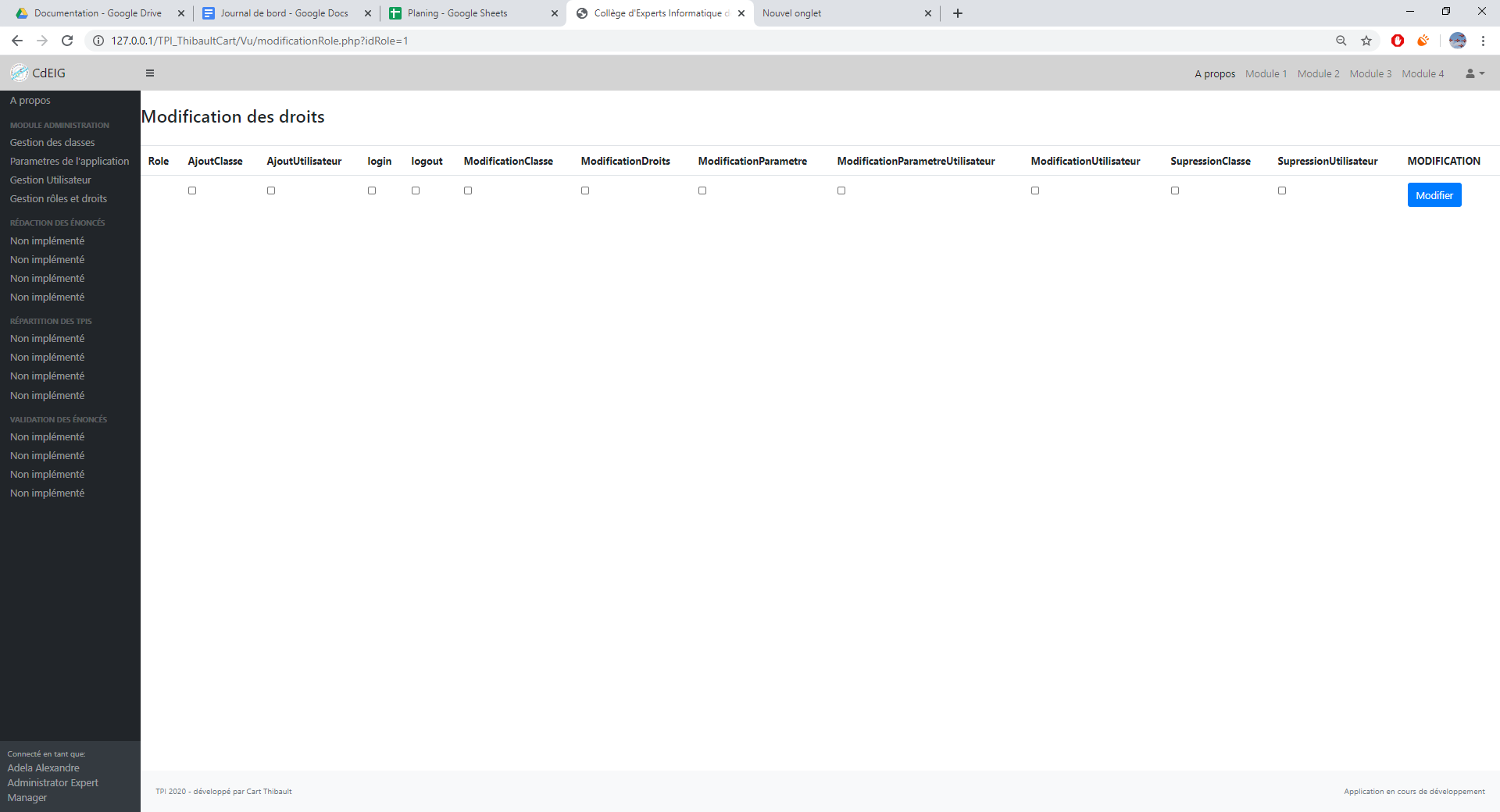
**Page Gestion rôles et droits**



Page Gestion des rôles et droits

Cette page permet de visualiser les rôles ainsi que leurs droits. Un appui sur le bouton “Modifier” nous redirige vers une autre page qui nous permettra de modifier les droits des utilisateurs.

**Page modification rôles**



Page Modification rôles

Page permettant de modifier les droits accordés à un rôle via des checkboxs.Les données sont traitées afin d’éviter les injections de code.

# Analyse organique

## Architecture du site

J’ai séparé mon code en trois parties principales qui sont: la vue, le modèle et le contrôleur.

**Vue:** Affichage du site, via les actions effectuées sur les pages, nous appelons des fonctions du contrôleur. Nous faisons parfois appel au modèle directement depuis la vue afin d’obtenir des données à afficher.

**Contrôleur:** Cerveau du site Web, tout ce qui est logique se passe ici. Les données saisies dans la vue sont récupérées et traitées ici. Une fois les données récupérées, filtrées et traitées, le contrôleur les transmet à la base de données via des fonctions présentes dans le modèle.

**Modèle:** Le modèle s’occupe de tout ce qui est relations avec la base de données, par le biais de requête. Il modifie, supprime, ajoute, récupère des données dans la base.

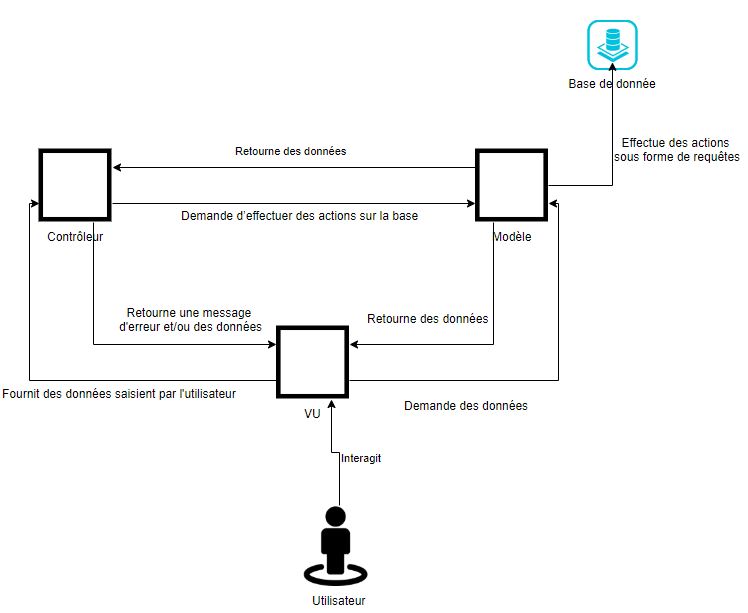


Image montrant le fonctionnement du site et des relations entre le modèle,la vu et le contrôleur

## Sécurité:

**Données stockées dans la base:**

Après analyse de la base de données, j’en suis arrivé à la conclusion que les seules données sensibles présentes dans la base de données sont: le mot de passe, le sel qui permet de crypter le mot de passe.

C’est pourquoi j’ai défini que le sel est un nombre aléatoire de 10 caractères cryptés avec l’algorithme SHA-1. Le mot de passe quant à lui est une concaténation du mot de passe saisi par l’utilisateur et du sel. Le résultat de cette concaténation est à son tour crypté avec l’algorithme SHA-1.

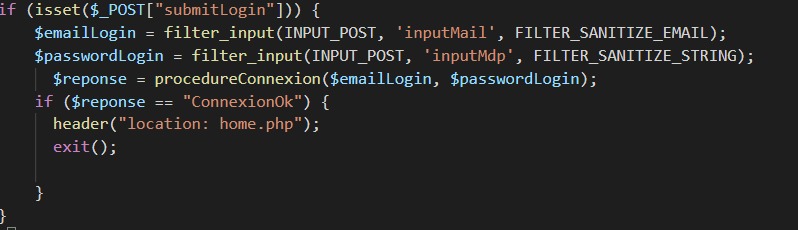
Je tiens à préciser que le SHA-1 n’est pas un choix personnel, la base de données fournie fonctionnait déjà avec ce système. C’est pourquoi dans un avenir proche, il faudrait changer de méthode de cryptage avec par exemple la fonction Password Hash de PHP.

**Informations saisies par l’utilisateur:**

Les informations saisies par l’utilisateur sont filtrées avec des filter\_input() afin d’éviter les injections SQL et XSS.

De plus, certaines valeurs sont validées grâce la fonction php FILTER\_VALIDATE\_EMAIL.

Grâce à cette fonction, si le mail n'est pas valide nous le savons et pouvons afficher un message d’erreur.



extrait de code de la page login.php

## Base de données:

La base de données a été créer par M: Comminot et contient 18 tables.

**MCD (Modèle Conceptuel des Données):**

Pour des raisons de clarté, je n’affiche que les tables que j’ai utilisées dans ce travail.

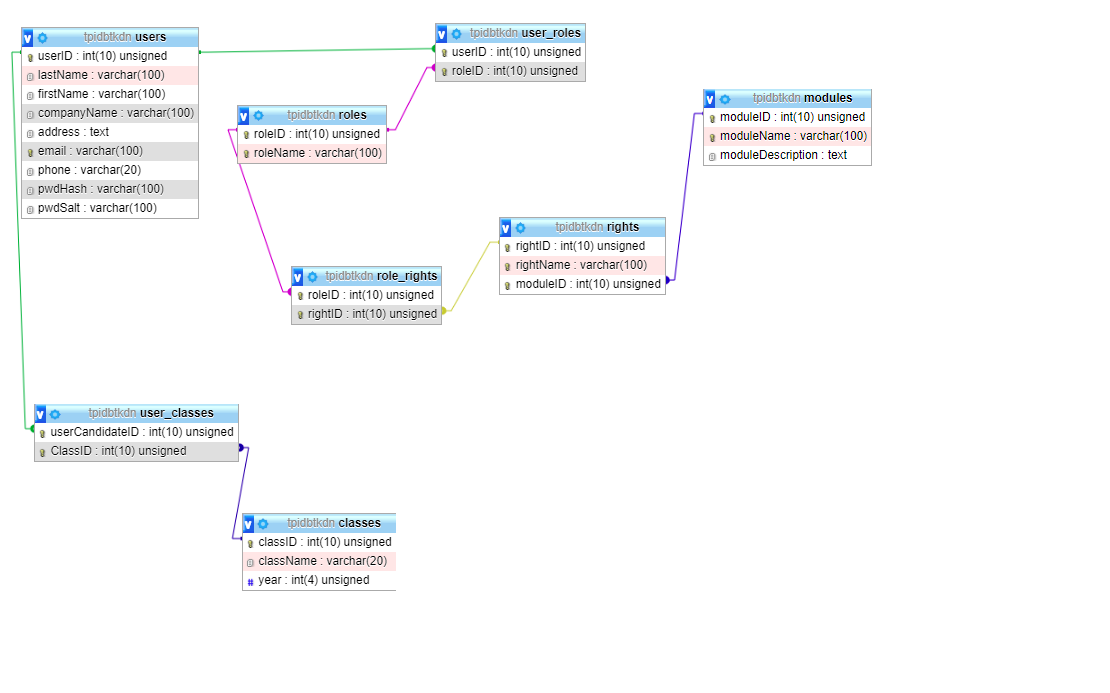


image montrant la base de données dans un format MCD

# Protocole et rapport de test

Dans ce chapitre, je vais décrire les différents tests effectués afin de garantir un bon fonctionnement du site.

**Login**

|  |  |  |  |
| --- | --- | --- | --- |
| **N° Test** | **Description test** | **Résultat voulu** | **Validé** |
| 1 | Lors ce qu’on arrive sur la page. | La page s’affiche correctement. | Oui |
| 2 | Lors de l’appui sur le bouton “Se connecter”. | Le formulaire est envoyé. | Oui |
| 3 | Les champs sont protégés contre l’injection de code. | Les champs sont filtrés et validés. | Oui |
| 4 | Nous rentrons des informations de connexion correctes. | Nos données sont chargées en Session et nous somme redirigés vers la page Home. | Oui |

**Home**

|  |  |  |  |
| --- | --- | --- | --- |
| **N° Test** | **Description test** | **Résultat voulu** | **Validé** |
| 1 | Nous arrivons sur la page en n’étant pas connectés. | Nous redirige vers la page login. | Oui |
| 2 | La page s’affiche correctement. | La page s’affiche correctement. | Oui |
| 3 | Nos information personnelles sont bien affichées en bas de la navigation verticale. | Le prénom et le nom ainsi que les rôles de l’utilisateur connecté sont affichés en bas de la navigation verticale. | Oui |
| 4 | Le contenu de la navigation s’affiche dynamiquement en fonction des rôles de l’utilisateur. | Si l’utilisateur est administrateur alors la navigation lui montre le module administration. | Oui |
| 5 | Appui sur le bouton déconnexion sur la navigation horizontale. | Nous sommes déconnectés et toutes les données en session sont détruites. | Oui |
| 6 | Les boutons de la navigation redirige vers les bonnes pages. | Nous sommes redirigés vers la page correspondante au bouton cliqué. | Oui |

**Paramètres**

|  |  |  |  |
| --- | --- | --- | --- |
| **N° Test** | **Description test** | **Résultat voulu** | **Validé** |
| 1 | Nous arrivons sur la page en n’étant pas connectés. | Nous redirige vers la page login. | Oui |
| 2 | La page s’affiche correctement. | La page s’affiche correctement. | Oui |
| 3 | Nos information personnelles sont affichées. | Nos informations personnelles ainsi que nos rôles sont affichés | Oui |
| 4 | Nous cliquons sur le bouton “Modifier mon mot de passe”. | Un formulaire modal s’ouvre afin que nous saisissons notre nouveau mot de passe. | Oui |
| 5 | Les formulaires sont protégés contre les injections de code. | Si je rentre du code dans les champs du formulaire, ceci n’aura pas d’incidence car je filtre et valide les champs auparavant. . | Oui |

**Gestion des classes**

|  |  |  |  |
| --- | --- | --- | --- |
| **N° Test** | **Description test** | **Résultat Voulu** | **Validé** |
| 1 | Nous arrivons sur la page en n’étant pas connectés. | Nous redirige vers la page login. | Oui |
| 2 | La page s’affiche correctement. | La page s’affiche correctement. | Oui |
| 3 | Nous pouvons accéder à cette page seulement si nous sommes administrateurs. | Si nous ne sommes pas administrateurs, nous redirige vers la page home. | Oui |
| 4 | Les fonctionnalités s'affichent selon nos droits | Selon nos droits et nos rôles nous pouvons voir et utiliser différentes fonctionnalités de la page. | Oui |
| 5 | L’affichage des classes est fonctionnel. | La page nous affiche les classes présentes dans la base de données ainsi que le nombre d'élèves présents dans chaque classe. | Oui |
| 6 | L’appui sur le bouton “Supprimer” supprime la classe de la base de données à condition qu’elle soit vide. | La classe est supprimée à condition qu'elle soit vide. | Oui |
| 7 | L’appui sur le bouton “Modifier” nous redirige vers une page qui nous permettra de modifier la classe. | Le bouton “Modifier” nous redirige vers la page “modificationClasse” afin de pouvoir modifier la classe, l’id de la classe est transmise en paramètre dans l’URL. | Oui |
| 8 | L’appui sur le bouton “Ajouter une classe” ouvre une formulaire modal qui nous permet d’ajouter un classe. | L’appui sur le bouton “Ajouter une classe” affiche le formulaire qui permet d’ajouter une classe et l’ajoute. | Oui |
| 9 | Le formulaire est protégé contre les injections de code. | Si je rentre du code dans les champs du formulaire, ceci n’aura pas d’incident car je filtre et valide les champs avant. | Oui |

**Modification classe**

|  |  |  |  |
| --- | --- | --- | --- |
| **N° Test** | **Description test** | **Résultat Voulu** | **Validé** |
| 1 | Nous arrivons sur la page en n’étant pas connectés. | Nous redirige vers la page login. | Oui |
| 2 | La page s’affiche correctement | La page s’affiche correctement. | Oui |
| 3 | Nous pouvons accéder à cette page seulement si nous sommes administrateurs. | Si nous ne somme pas administrateurs nous redirige vers la page home. | Oui |
| 4 | Les fonctionnalités s'affichent selon nos droits. | Selon nos droits et nos rôles nous pouvons voir et utiliser différentes fonctionnalitées de la page | Oui |
| 5 | Affiche le précédent nom de la classe en arrière plan dans le champ du formulaire. | Grâce à l’id de la classe transmis via l’URL, récupère l’ancien nom de la classe dans la base de donnée et l’affiche dans le champ du formulaire. | Oui |
| 6 | Le formulaire s'envoie et modifie le nom de la classe. | Le formulaire s’envoie, les données sont vérifiées et modifiées dans la base de données. | Oui |
| 7 | Le formulaire est protégé contre les injections de code. | Si je rentre du code dans les champs du formulaire, ceci n’aura pas d'incidence car je filtre et valide les champs avant. | Oui |

**Suppression classe**

|  |  |  |  |
| --- | --- | --- | --- |
| **N° Test** | **Description test** | **Résultat Voulu** | **Validé** |
| 1 | Nous arrivons sur la page en n’étant pas connectés. | Nous redirige vers la page login. | Oui |
| 2 | La page s’affiche correctement. | La page s’affiche correctement. | Oui |
| 3 | Nous pouvons accéder à cette page seulement si nous sommes Administrateur | Si nous ne somme pas Administrateur, nous redirige vers la page Home. | Oui |
| 4 | Les fonctionnalités s'affichent selon nos droits. | Selon nos droits et nos rôles nous pouvons voir et utiliser différentes fonctionnalitées de la page. | Oui |
| 5 | Vérifie si la classe est vide. | Avec l’id de la classe reçue en paramètre dans l’URL vérifie si la classe est vide. | Oui |
| 6 | Supprime la classe. | Si la classe est vide alors nous la supprimons de la base. | Oui |

**Paramètres de l'application**

|  |  |  |  |
| --- | --- | --- | --- |
| **N° Test** | **Description test** | **Résultat Voulu** | **Validé** |
| 1 | Nous arrivons sur la page en n’étant pas connectés | Nous redirige vers la page login. | Oui |
| 2 | La page s’affiche correctement | La page s’affiche correctement | Oui |
| 3 | Nous pouvons accéder à cette page seulement si nous sommes Administrateur. | Si nous ne somme pas Administrateur, nous redirige vers la page Home. | Oui |
| 4 | Les fonctionnalités s'affichent selon nos droits | Selon nos droit et nos rôles nous pouvons voir et utiliser différentes fonctionnalitées de la page. | Oui |
| 5 | Les paramètres de l’application sont affichés. | Les paramètres de l’application sont recherchés dans la base de données et affichés. | Oui |
| 6 | Le bouton “Modifier les paramètres” ouvre un formulaire modal qui permettra de modifier les données de l’application. | Le formulaire modal s’ouvre et permet de changer les paramètres de l'application, Si le formulaire est envoyé et que les données sont valides, modifie les paramètres du site. | Oui |
| 7 | Le formulaire est protégé contre les injections de code. | Si je rentre du code dans les champs du formulaire, cela n’aura pas d’incidence, car je filtre et valide les champs avant. | Oui |

**Gestion rôles et Droit**

|  |  |  |  |
| --- | --- | --- | --- |
| **N° Test** | **Description test** | **Résultat Voulu** | **Validé** |
| 1 | Nous arrivons sur la page en n’étant pas connectés. | Nous redirige vers la page login. | Oui |
| 2 | La page s’affiche correctement. | La page s’affiche correctement. | Oui |
| 3 | Nous pouvons accéder à cette page seulement si nous sommes Administrateur. | Si nous ne somme pas Administrateur, nous redirige vers la page Home. | Oui |
| 4 | Les fonctionnalités s'affichent selon nos droits. | Selon nos droit et nos rôles, nous pouvons voir et utiliser différentes fonctionnalités de la page. | Oui |
| 5 | Le tableau qui montre les différents rôles et les droits qui sont attribués à chaque rôle s’affiche. | Les rôles et les droits sont chargés et affichés dynamiquement. | Oui |
| 6 | L’appui sur le bouton “Modifier” nous redirige vers une autre page qui nous permettra de modifier les droits liés à ce rôle. | L’appui sur le bouton “Modifier” nous redirige vers une autre page qui nous permettra de modifier les droits liés aux rôles, l’id du rôle à modifier est transmise via les paramètres de l’URL. | Oui |

**Modification Droits**

|  |  |  |  |
| --- | --- | --- | --- |
| **N° Test** | **Description test** | **Résultat Voulu** | **Validé** |
| 1 | Nous arrivons sur la page en n’étant pas connectés. | Nous redirige vers la page login. | Oui |
| 2 | La page s’affiche correctement. | La page s’affiche correctement. | Oui |
| 3 | Nous pouvons accéder à cette page seulement si nous sommes Administrateur. | Si nous ne somme pas Administrateur, nous redirige vers la page Home. | Oui |
| 4 | Les fonctionnalités s'affichent selon nos droits. | Selon nos droits et nos rôles, nous pouvons voir et utiliser différentes fonctionnalitées de la page. | Oui |
| 5 | Le tableau affiche le rôle et les droits qui lui sont attribués. | Le rôle et les droits sont chargés et affichés dynamiquement. | Oui |
| 6 | L’appui sur le bouton “Modifier” modifie les droits reliés à ce rôle. | L’appui sur le bouton “Modifier” modifie les droits reliés à ce rôle. | Oui |
| 7 | Le formulaire est protégé contre les injections de code. | Si je rentre du code dans les champs du formulaire, cela n’aura pas d’incidence car je filtre et valide les champs avant. | Oui |

**Gestion utilisateurs**

|  |  |  |  |
| --- | --- | --- | --- |
| **N° Test** | **Description test** | **Résultat Voulu** | **Validé** |
| 1 | Nous arrivons sur la page en n’étant pas connectés. | Nous redirige vers la page login. | Oui |
| 2 | La page s’affiche correctement. | La page s’affiche correctement. | Oui |
| 3 | Nous pouvons accéder à cette page seulement si nous sommes Administrateur. | Si nous ne somme pas Administrateur, nous redirige vers la page Home. | Oui |
| 4 | Les fonctionnalités s'affichent selon nos droits | Selon nos droit et nos rôles nous pouvons voir et utiliser différentes fonctionnalitées de la page. | Oui |
| 5 | L’affichage des utilisateurs  est fonctionnel. | La page nous affiche les utilisateurs présents dans la base de données ainsi que leurs rôles. | Oui |
| 6 | L’appui sur le bouton “Supprimer” supprime l’utilisateur de la base de données à condition qu’il ne soit relié à aucun TPI. | L’utilisateur est supprimé de la base de données à condition qu'il ne soit relié à aucun TPI. | Oui |
| 7 | L’appui sur le bouton “Modifier” nous redirige vers une page qui nous permettra de modifier l’utilisateur. | Le bouton “Modifier” nous redirige vers la page “modification utilisateur” afin de pouvoir modifier l’utilisateur, l’id de l'utilisateur est transmise en paramètre dans l’URL. | Oui |
| 8 | L’appui sur le bouton “Ajouter un utilisateur” ouvre une formulaire modal qui nous permet de créer un utilisateur. | L’appui sur le bouton “Ajouter une utilisateur” affiche le formulaire qui permet d’ajouter un utilisateur et l’ajoute à la base de données si les informations saisie sont correctes. | Oui |
| 9 | Le formulaire est protégé contre les injections de code. | Si je rentre du code dans les champs du formulaire, cela n’aura pas d’incidence car je filtre et valide les champs avant . | Oui |

## Conclusion rapport de test

Une fois tous les tests effectués, nous pouvons donc voir que mon application est fonctionnelle. De plus j’ai fait le maximum à ma connaissance pour réaliser une application sécurisée.

Il n’existe aucune faille de sécurité, du moins je ne connais personnellement aucun façon de porter atteinte à l'intégrité du site avec les sécurités misent en place.

# Conclusion

Ce TPI a constitué pour moi le point final de ma formation.

A cette occasion, j'ai pu utiliser toutes les connaissances apprises lors de mon CFC. Ce TPI est le plus gros projet personnel que j’ai fait en informatique. C’était pour moi l’occasion de de revoir mes bases et utiliser des connaissances datant du début de ma formation. Je pense que lors de mon TPI, c’est en SQL que j’ai le plus appris. C’est un langage que je redoutais un peu au début, or je me suis au final très bien débrouillé et même au-delà de mes attentes.

D’un point de vue sécurité j’ai mis en pratique tous les enseignements appris en cours afin d’obtenir un site le plus sûr possible. Toutefois, l’utilisation de l’algorithme Sha-1 me chiffonne. En effet cet algorithme se fait vieillissant et l’utilisation de la fonction Password HASH me semble plus adaptée à la situation, en plus de pouvoir supprimer l'utilisation du salage de mot de passe.

Pour conclure, je suis plutôt satisfait de mon TIP, en effet, même si j’ai consacré trop peu de temps à la documentation technique, j’ai quand-même pu terminer l’écriture de la documentation technique à temps sur les derniers jours.

Ce projet a été réalisé dans des conditions particulières liées au COVID, et j'ai dû gérer un problème imprévu. A cause de travaux sur la toiture de mon immeuble, et du bruit que ça occasionnait, j’ai pris du retard sur le planning. Heureusement j’ai pu terminer mon TPI au CFPT, ce qui m’a permis de travailler dans le calme, et rattrapper une bonne bonne partie de mon retard. Toutefois, il m’a juste manqué le temps d’inclure des messages d’erreurs sur toutes les pages du site. Heureusement, ceci ne pose pas de problème de fonctionnement. En revanche, d’un point de vue ergonomique, ces messages d’erreurs auraient été souhaitables. Enfin, j’aurai volontiers amélioré le design de certaines pages, si j’avais eu du temps supplémentaire.

Ce TPI reste pour moi une expérience très positive et enrichissante, et j’ai apprécié me consacrer à ce projet. Je me réjouis de pouvoir réaliser d’autres projets d’envergure dans ma future carrière d’informaticien.

# 

# Annexes

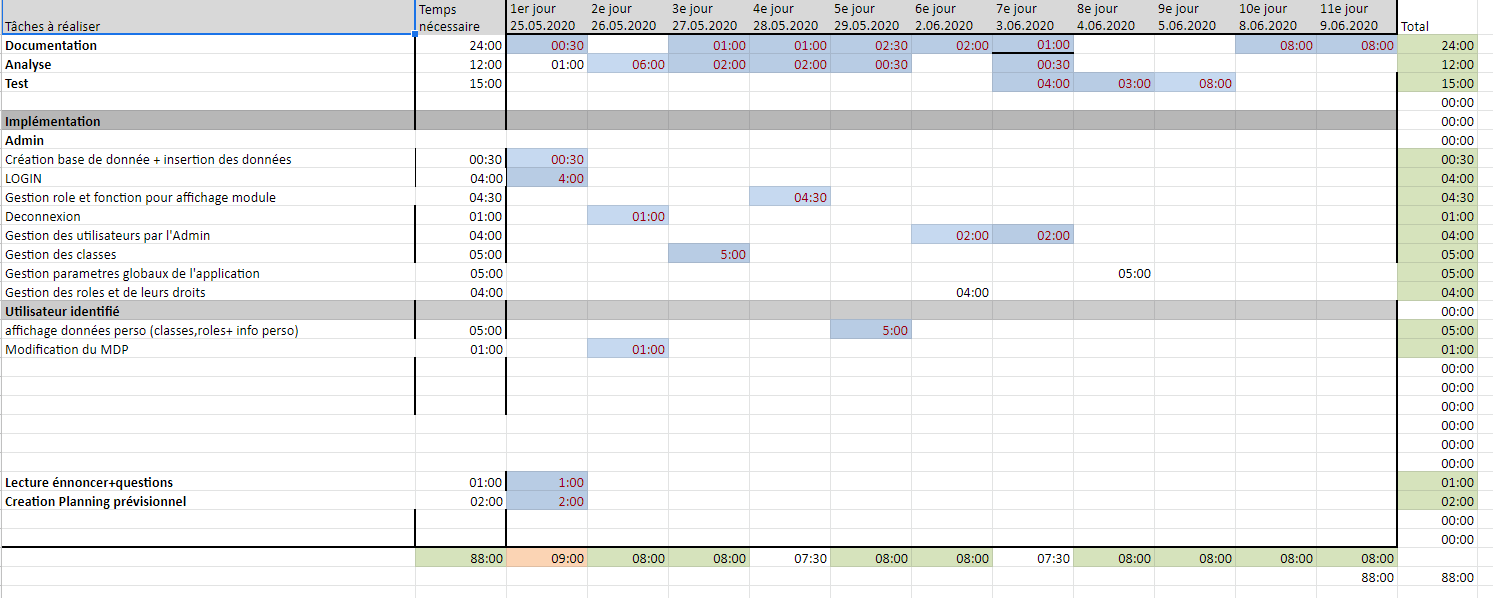
## Mode d’emploi

## Code Source

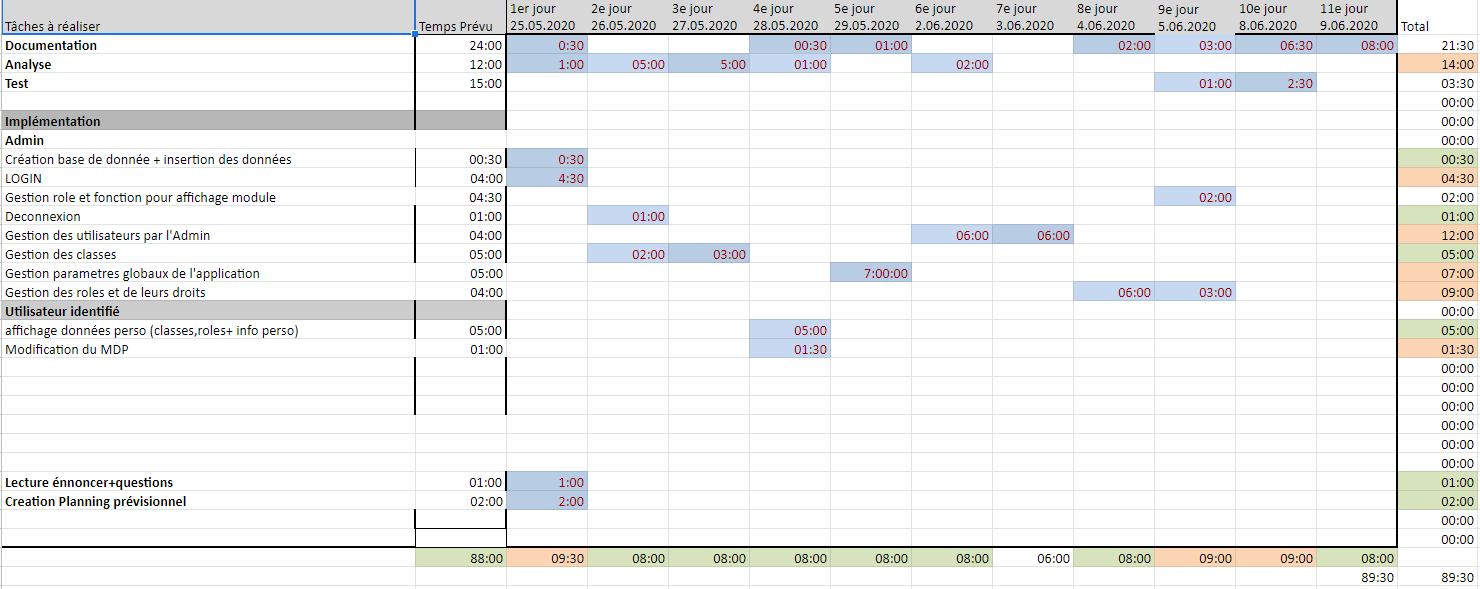
## Journal de bord

## Planning initial et final

**Planning initial**



**Planning final**

****

## 

## Bibliographie et webographie

Utilisée afin de faire la déconnexion du site: <https://www.php.net/manual/fr/function.session-destroy.php>

Utilisée pour le design des pages:<https://getbootstrap.com/docs/4.5/components/forms/>

## Glossaire

Piratage par force brute: Essai de beaucoup de combinaisons d’adresses mails et de mots de passe afin trouver un compte dans la base de données.

IDE: Integrated Development Environment

autocomplétion: le programme vous propose en rapport avec ce que vous tapez de finir le mots

SQL et XSS: le SQL est un langage permettant d'effectuer des actions sur une base de données.Le XSS plus connus sous le nom de cross-site-scripting est un act malveillant d’injecter du code qui va s’exécuter côté clients et effectuer différentes actions malveillantes.