09/06/2020

Adrien Witzig

CFPT-I

Evaluation EE

Documentation technique

# Table des matières

[1 Table des matières 2](#_Toc42541401)

[2 Introduction 4](#_Toc42541402)

[3 Cahier des charges 4](#_Toc42541403)

[3.1 Objectifs 4](#_Toc42541404)

[3.2 Description détaillée du projet 4](#_Toc42541405)

[3.3 Organisation 5](#_Toc42541406)

[3.4 Environnement 5](#_Toc42541407)

[3.5 Modèle conceptuel de données 5](#_Toc42541408)

[3.6 Planning prévisionnel 6](#_Toc42541409)

[4 Analyse fonctionnelle 6](#_Toc42541410)

[4.1 Technologies 6](#_Toc42541411)

[4.2 Connexion utilisateur 7](#_Toc42541412)

[4.3 Sécurité d’authentification à travers les pages 7](#_Toc42541413)

[4.4 Fonctionnalités par statut d’utilisateur 7](#_Toc42541414)

[4.4.1 Utilisateur « Elève » 7](#_Toc42541415)

[4.4.2 Utilisateur « Enseignant » 7](#_Toc42541416)

[4.4.3 Utilisateur « Administrateur » 7](#_Toc42541417)

[4.5 Maquettes du site 7](#_Toc42541418)

[5 Analyse organique 23](#_Toc42541419)

[5.1 Architecture du code 23](#_Toc42541420)

[5.2 Arborescence de fichiers 23](#_Toc42541421)

[5.3 Organigramme des pages 24](#_Toc42541422)

[5.4 Implémentation de la sécurité entre les pages 24](#_Toc42541423)

[5.5 Librairie FPDF 25](#_Toc42541424)

[5.6 Librairie Bootstrap 25](#_Toc42541425)

[5.7 JavaScript et libraire JQuery 26](#_Toc42541426)

[5.8 Evaluation d’un élève 26](#_Toc42541427)

[5.9 Fonctions et scripts PHP principaux 27](#_Toc42541428)

[5.9.1 *connectDB()* 27](#_Toc42541429)

[*5.9.2* *getNote($idEleve, $idProjet)* 27](#_Toc42541430)

[*5.9.3* *checkLogin.php* 27](#_Toc42541431)

[*5.9.4* *estEvalue($idEleve, $idProjet)* 27](#_Toc42541432)

[*5.9.5* *getEleveToAdd($idProjet)* 27](#_Toc42541433)

[*5.9.6* *checkProjetExist($idProjet)* 28](#_Toc42541434)

[*5.9.7* *evaluer.php* 28](#_Toc42541435)

[*5.9.8* *changeMotDePasse.php* 28](#_Toc42541436)

[*5.9.9* *logout.php* 28](#_Toc42541437)

[5.9.10 Scripts d’ajouts 28](#_Toc42541438)

[5.9.11 Scripts de modification 29](#_Toc42541439)

[5.9.12 Scripts de suppression 29](#_Toc42541440)

[5.10 Modèle logique de données 30](#_Toc42541441)

[5.11 Suppressions de données 30](#_Toc42541442)

[5.12 Tables de données 31](#_Toc42541443)

[5.12.1 Table UTILISATEURS 31](#_Toc42541444)

[5.12.2 Table ELEVES 32](#_Toc42541445)

[5.12.3 Table TRAVAILLE\_POUR 32](#_Toc42541446)

[5.12.4 Table PROJETS 33](#_Toc42541447)

[5.12.5 Table DOMAINES 33](#_Toc42541448)

[5.12.6 Table CRITERES\_PROJETS 33](#_Toc42541449)

[5.12.7 Table CRITERES 34](#_Toc42541450)

[5.12.8 Table CATEGORIES 34](#_Toc42541451)

[5.12.9 Table EVALUATIONS 34](#_Toc42541452)

[6 Tests 35](#_Toc42541453)

[7 Conclusion 39](#_Toc42541454)

[7.1 Installation du projet 39](#_Toc42541455)

[7.2 Problèmes rencontrés 39](#_Toc42541456)

[7.3 Améliorations possibles 41](#_Toc42541457)

[7.4 Comparaison planning effectif 41](#_Toc42541458)

[7.5 Documents à rendre 42](#_Toc42541459)

[8 Bibliographie 42](#_Toc42541460)

[8.1 Sites d’aide à la programmation 42](#_Toc42541461)

[8.2 Hébergement sauvegarde 42](#_Toc42541462)

[9 Table des illustrations 43](#_Toc42541463)

[10 Tableaux 43](#_Toc42541464)

[11 Annexes 44](#_Toc42541465)

[11.1 Manuel utilisateur 44](#_Toc42541466)

[11.2 Résumé du TPI 44](#_Toc42541467)

[11.3 Planification 44](#_Toc42541468)

[11.4 Journal de bord 44](#_Toc42541469)

# Introduction

Cette documentation démontre les étapes et procédés dans la conception du site Evaluation de projets EE. Je réalise ce projet à l’occasion du Travail Pratique Individuel (TPI). Ce site web a pour but de permettre à l’école entreprise d’informatique, d’avoir une plateforme où les enseignants pourraient noter les élèves dans leur projet via des critères et d’en développer une note. Pour les élèves, il s’agit de pouvoir consulter ses notes de projet ainsi que d’avoir la possibilité de télécharger directement son évaluation. L’application web n’a aucune fonctionnalités en tant que utilisateur non-authentifié. L’enseignant peut ajouter son projet à la plateforme, il devra ensuite y ajouter des élèves et enfin les noter.

# Cahier des charges

## Objectifs

L’objectif de ce site Web est de crée une plateforme de gestion des évaluations dans le cadre de l’école entreprise du CFPT. Il pourra profiter aux enseignants, pour qui l’évaluation sera plus organisée, mais aussi aux élèves qui pourront consulter et télécharger[[1]](#footnote-1) leurs notes.

## Description détaillée du projet

L’application ne contient aucune fonction d’utilisateur non-authentifié, il s’agit donc de se connecter. Les identifiants de connexion sont régis par l’administrateur. Les utilisateurs (élèves, enseignants et administrateurs) se connectent avec leurs noms, prénoms et mots de passes.

L’administrateur peut gérer (ajouter/modifier/supprimer) les utilisateurs. Il peut également gérer (ajouter/modifier/supprimer) les catégories des critères et les domaines de projets. Il peut également modifier et supprimer les projets.

Les élèves peuvent consulter leurs projets. En effet, si leur évaluation a été réalisée, ils peuvent la consultée et même la télécharger. Dans un projet, ils peuvent aussi voir leurs collaborateurs, sans voir s’il a été évalué.

Les enseignants peuvent ajouter leurs projets, puis lui assigner des élèves et par la suite évaluer les élèves en fonctions de critères établis pour ce projet. Les critères peuvent être ajoutés au projet parmi les critères existants. L’enseignant pourra également ajouter, modifier ou supprimer des critères.  
Les enseignants auront la possibilité de voir les évaluations qu’ils ont faite et les télécharger.

Chaque évaluation donne lieu à une note calculée par le barème fédéral (voir 27).

## Organisation

*Exécutant TPI :* Adrien Witzig

*Maitre d’apprentissage :* Sandrine Weber

*Experts :* Pascal Court et Borys Folomietow

## Environnement

* PC : Microsoft Surface
* Système d’exploitation : Windows 10, 64 bits
* Editeur de développement : Atom 1.46.0
* Serveur HTTP : EasyPhp Devserver 17.0 (Apache) en local
* Outil de gestion de base de données MySQL : PhpMyAdmin 4.7.0

## Modèle conceptuel de données

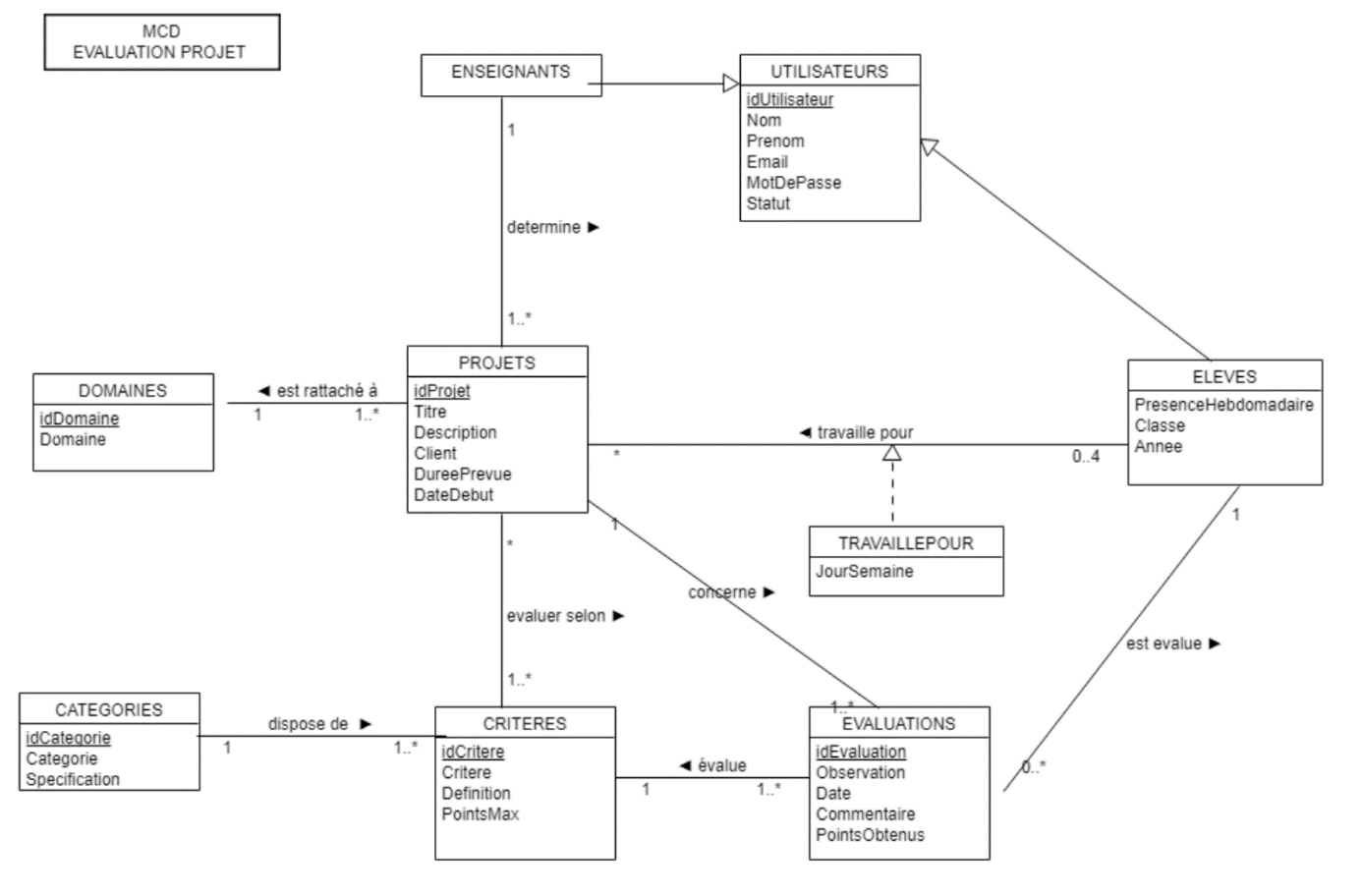


Figure 1 Modèle conceptuel de données fourni pour l'énoncé

## Planning prévisionnel

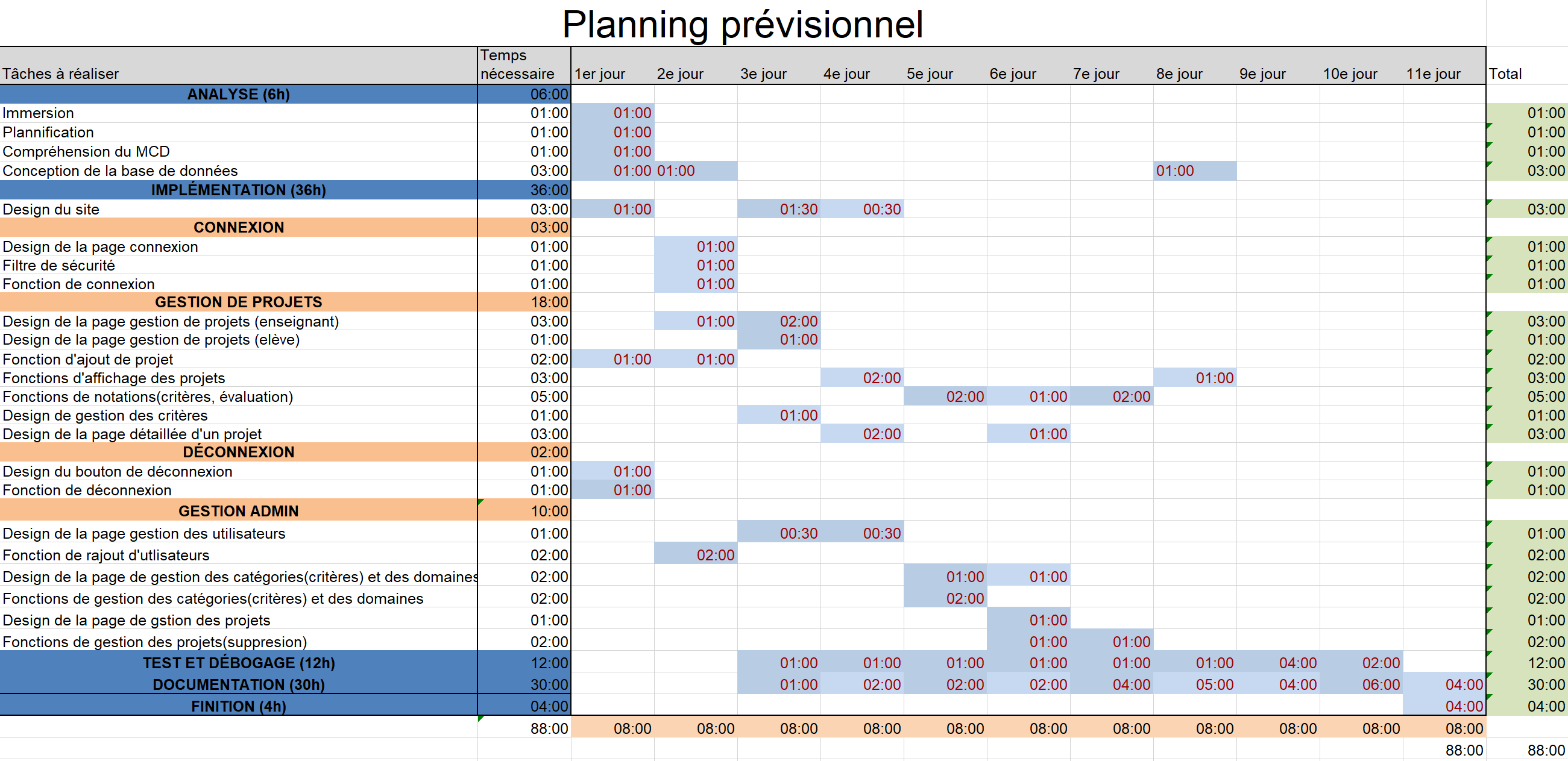


Figure Planning prévisionne77l7

# Analyse fonctionnelle

Il existe un grand nombre de site qui permettent de noter des élèves dans un travail de groupe, comme <http://classroom.google.com/> que nous utilisons fréquemment dans l’école. Ces sites sont très pratique mais ne sont pas assez adapté à l’école entreprise. En effet, ils ne permettent pas d’ajouter des critères d’évaluations et l’ajout d’élèves est laborieux car il faut souvent passer par l’invitation par email.

## Technologies

Le projet doit être un site Web, comme ça les élèves et les professeurs pourront y accéder plus facilement et depuis tous les appareils (ordinateurs, smartphones, tablettes, etc…).

J’utilise EasyPhp comme serveur HTTP ou de base de données.

La partie graphique est écrite en HTML/CSS et je me suis aidé de la librairie Bootstrap pour le CSS.

La partie logique, coté serveur, est écrite en PHP avec un petit peu de JavaScript.

## Connexion utilisateur

La connexion utilisateur se fait par le Prénom et Nom de l’utilisateur et le mot de passe. Lors d’un ajout d’utilisateur par l’administrateur, il peut arriver que deux personnes aient le même nom et prénom. Il faudra alors ajouter le deuxième prénom des utilisateurs ou, s’ils n’en ont pas, une spécificité comme la date de naissance. *Exemple* 🡪(*Prénom : John10.10.1998 / Nom : Doe)*

## Sécurité d’authentification à travers les pages

La sécurité d’accès des pages est gérée par les variables de sessions[[2]](#footnote-2) PHP.

## Fonctionnalités par statut d’utilisateur

Les fonctionnalités du site diffèrent selon le statut de l’utilisateur.

### Utilisateur « Elève »

* Se connecter avec son prénom et nom.
* Voir les projets auxquels il participe.
* Voir un détail du projet et ses collaborateurs (autres élèves associés)
* Voir son évaluation si elle a déjà été faite.
* Télécharger son évaluation en format .PDF

### Utilisateur « Enseignant »

* Se connecter avec son prénom et nom.
* Ajouter un projet.
* Voir les projets qu’il a créé.
* Voir un détail du projet, pouvoir le modifier et le supprimer.
* Ajouter/voir les élèves du projet
* Voir/Modifier/Ajouter/Supprimer des critères de manière générale.
* Ajouter ou supprimer des critères d’un projet
* Evaluer les élèves selon les critères.
* Télécharger l’évaluation d’un élève en format .PDF

### Utilisateur « Administrateur »

* Se connecter avec un prénom et nom.
* Voir/Modifier/Ajouter/Supprimer des critères de manière générale.
* Voir/Modifier/Ajouter/Supprimer des utilisateurs.
* Voir/Modifier/Ajouter/Supprimer des catégories de critères.
* Voir/Modifier/Ajouter/Supprimer des domaines de projets.
* Voir/Modifier/Ajouter/Supprimer des projets.

## Maquettes du site

Ces maquettes sont des captures d’écran des différentes vues du site.

Dans ce chapitre, je simplifie le terme fenêtre modal[[3]](#footnote-3) par Modal.



Figure 3 Fenêtre de connexion

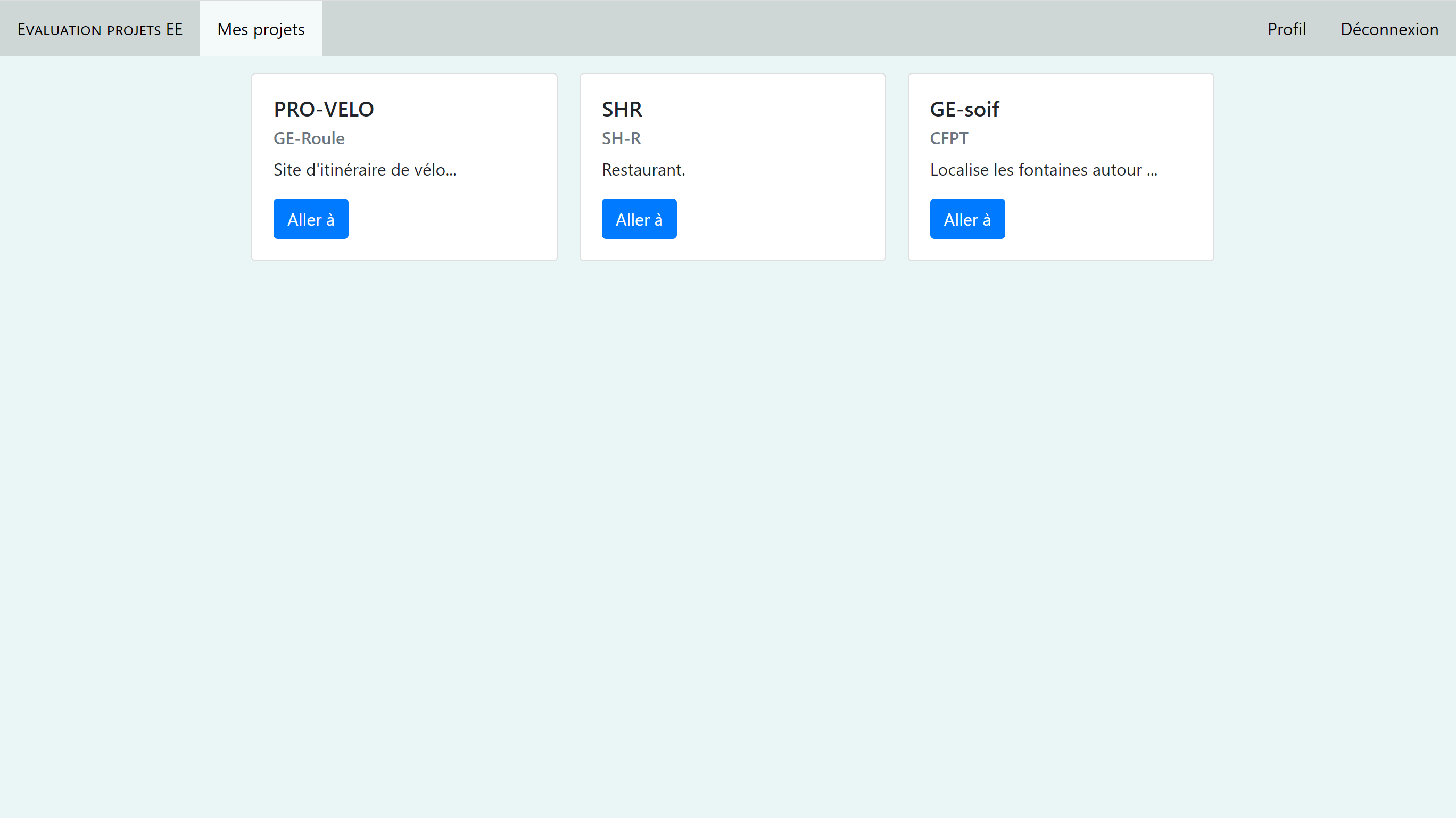


Figure 4 Page "Mes projets" (pour élèves)

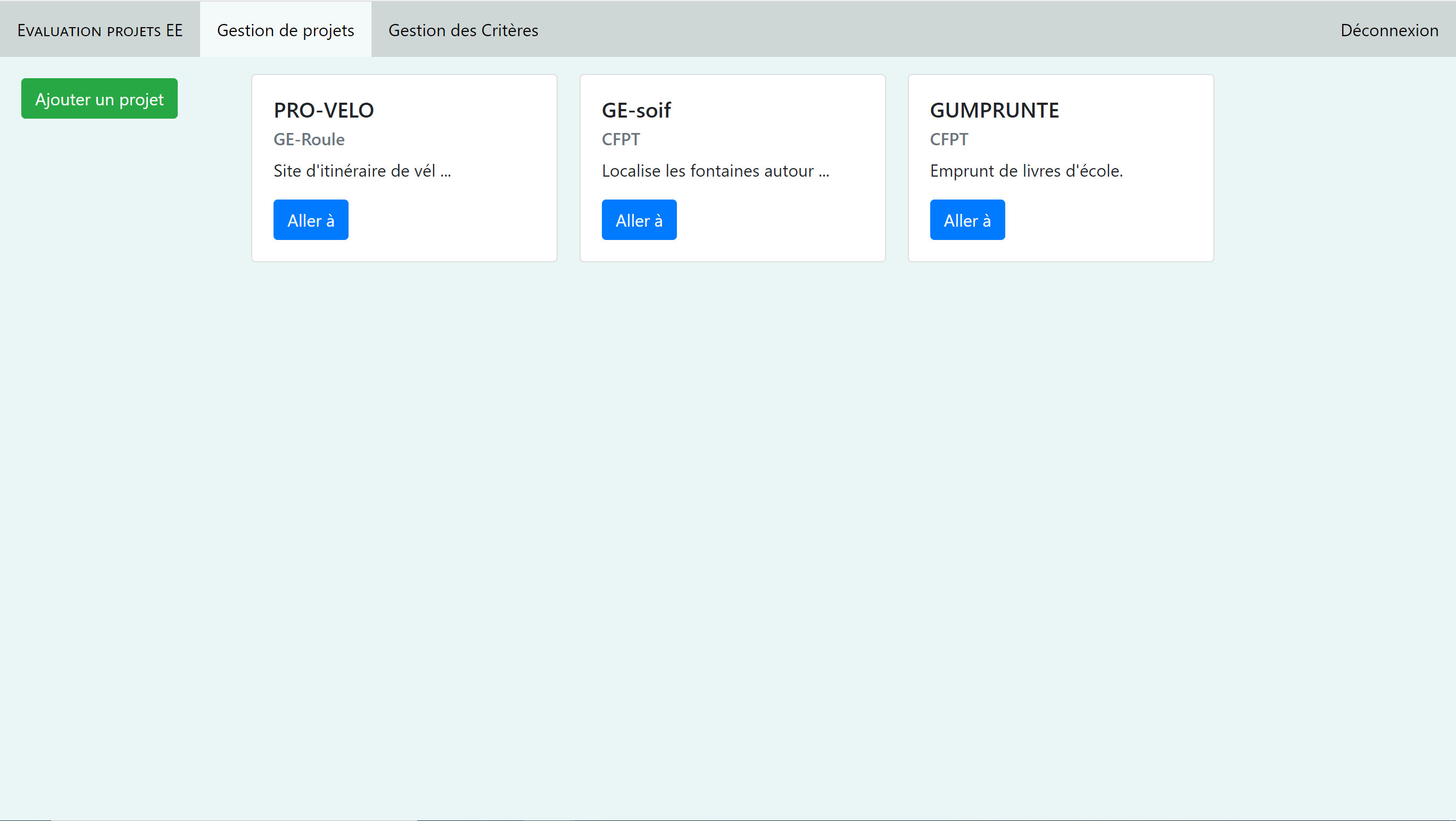


Figure 5 Page gestion de projets (pour enseignants)

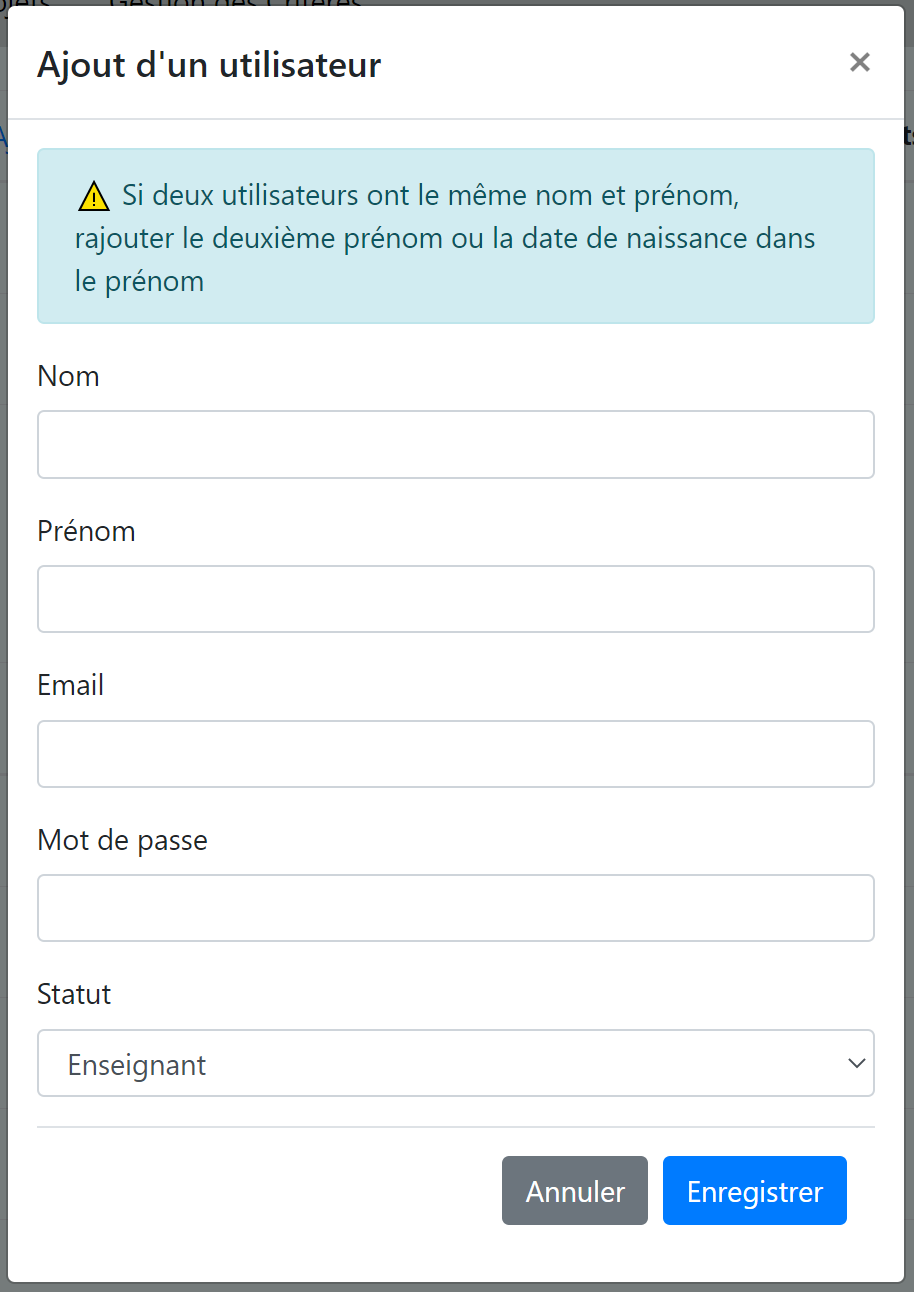


Figure 6 Modal d’ajout projets (pour enseignants)

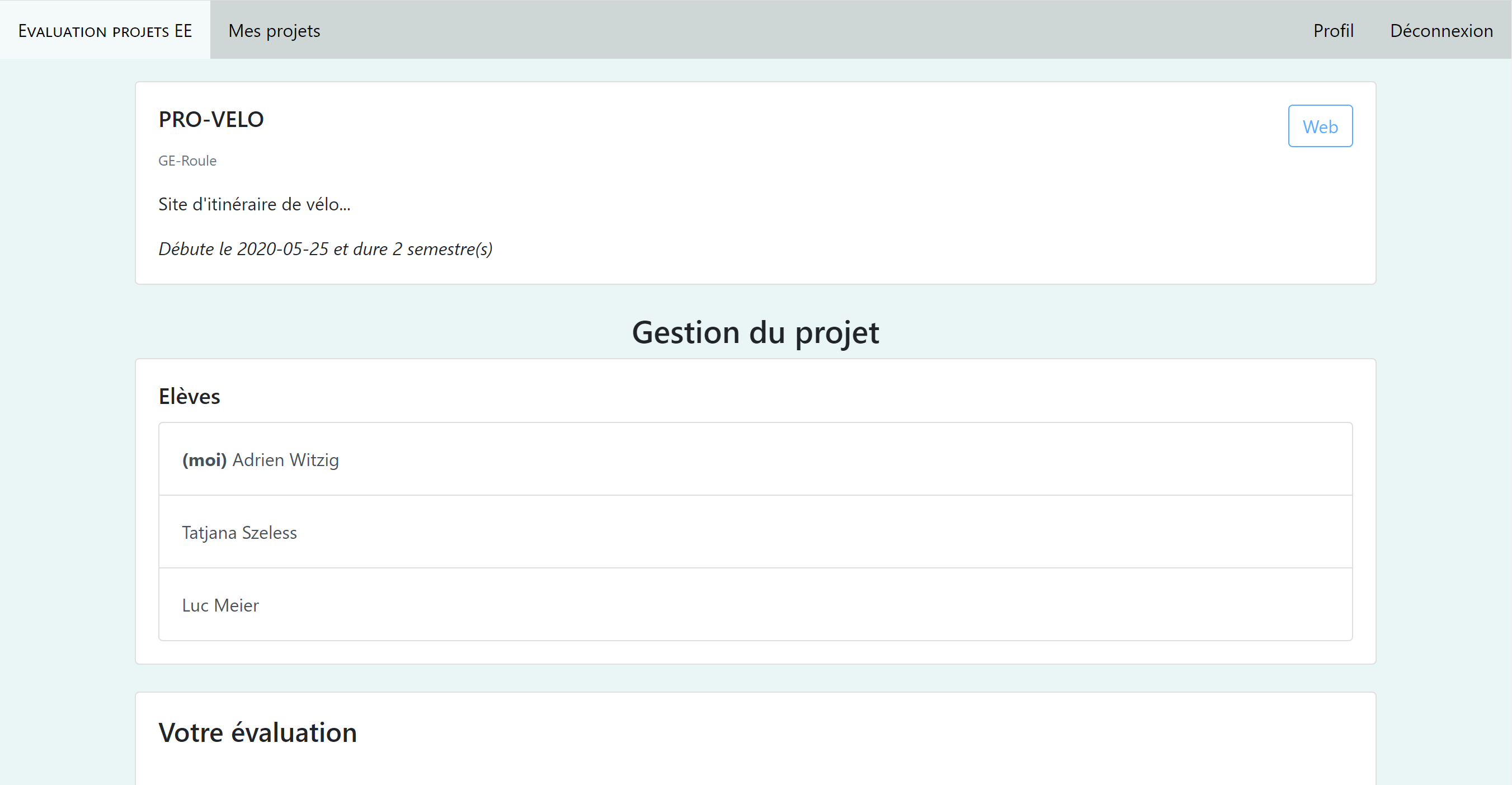


Figure 7 Page de détail d'un projet (pour élèves) partie 1

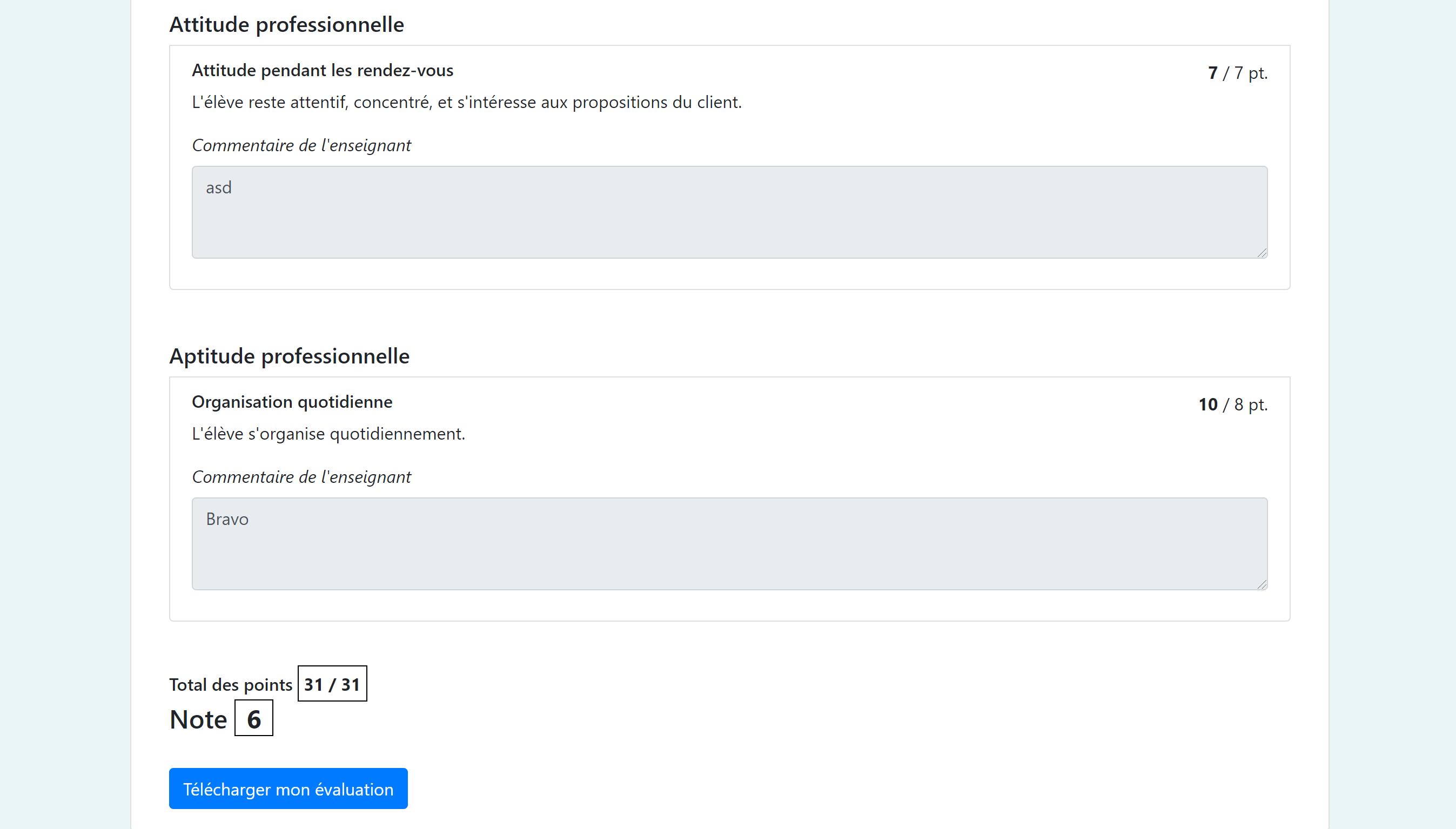


Figure 8 Page de détail d'un projet (pour élèves) partie 2

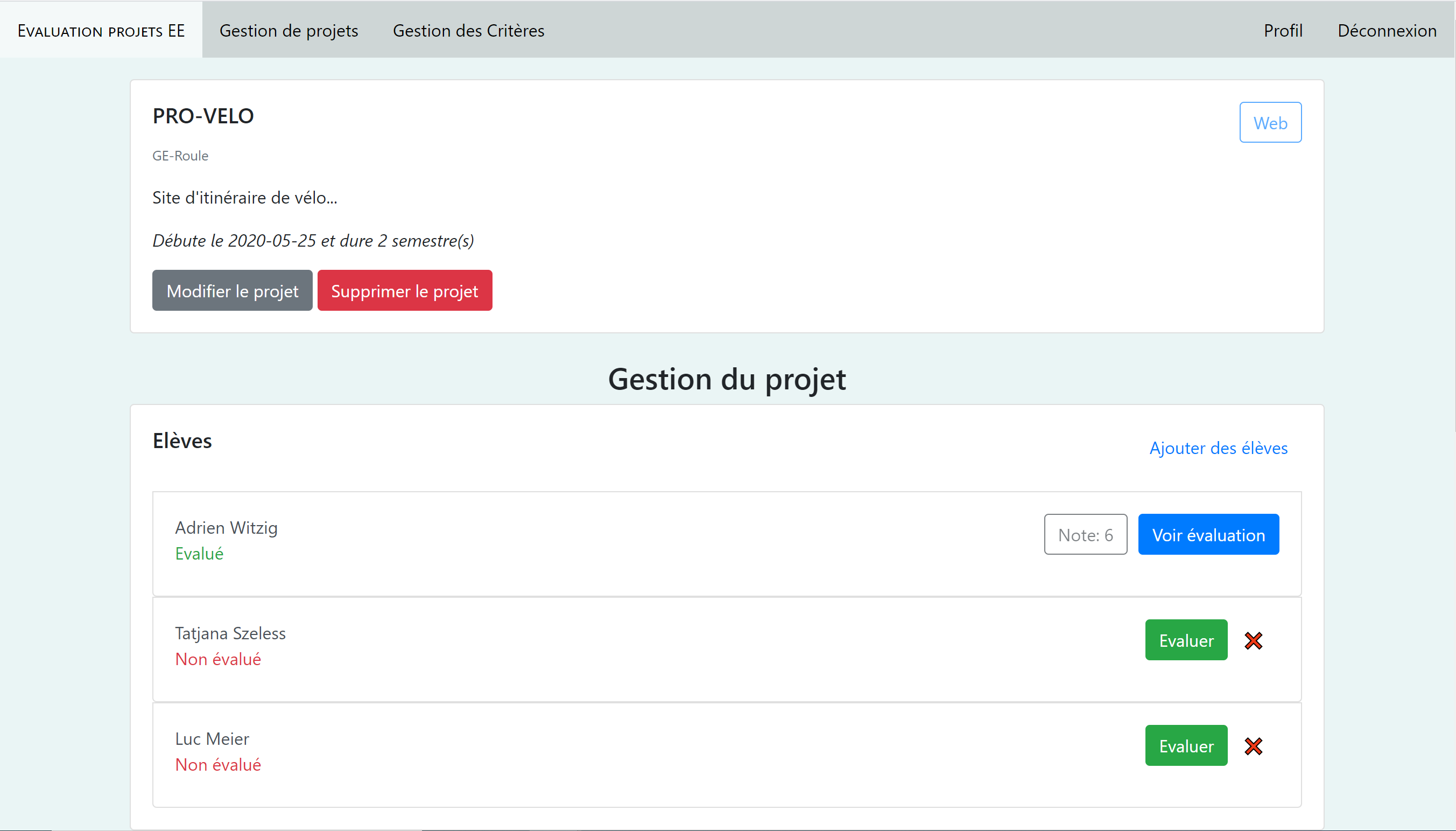


Figure 9 Page "gestion de projets" (pour enseignants) partie 1 avec élève évalué

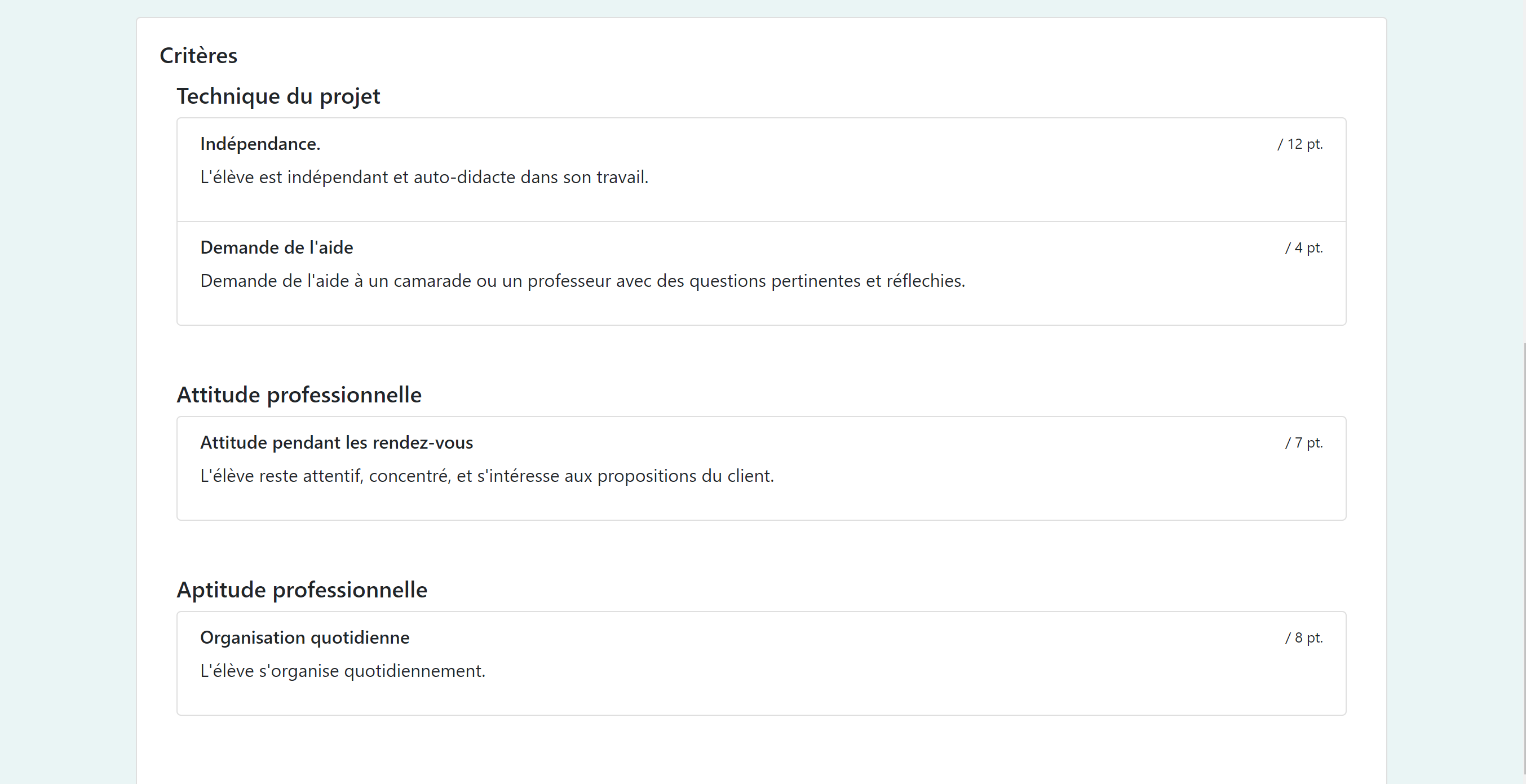


Figure 10 Page "gestion de projets" (pour enseignants) partie 2 avec élève évalué

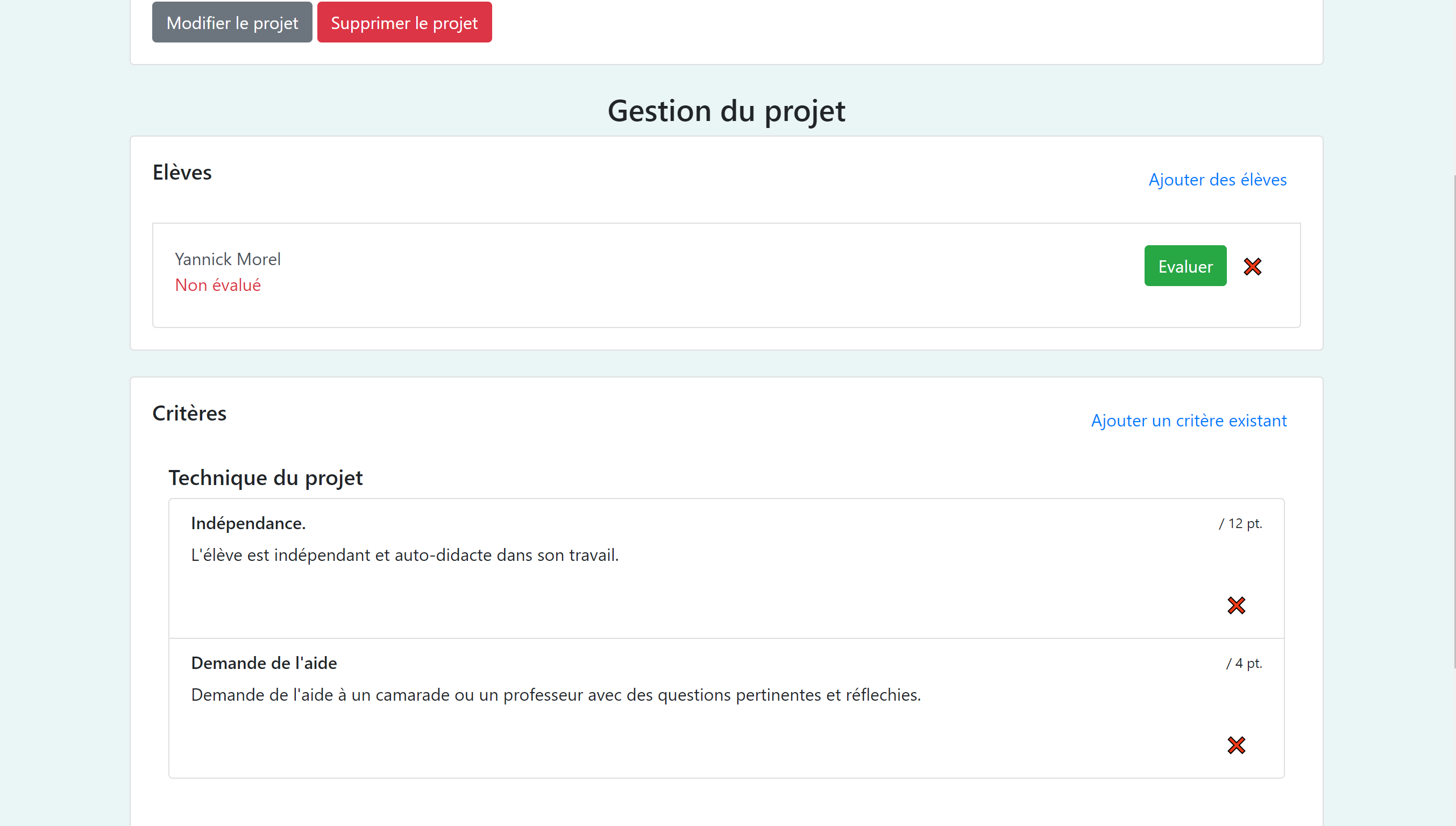


Figure 11 Page "gestion de projets" (pour enseignants) sans aucun élèves évalués

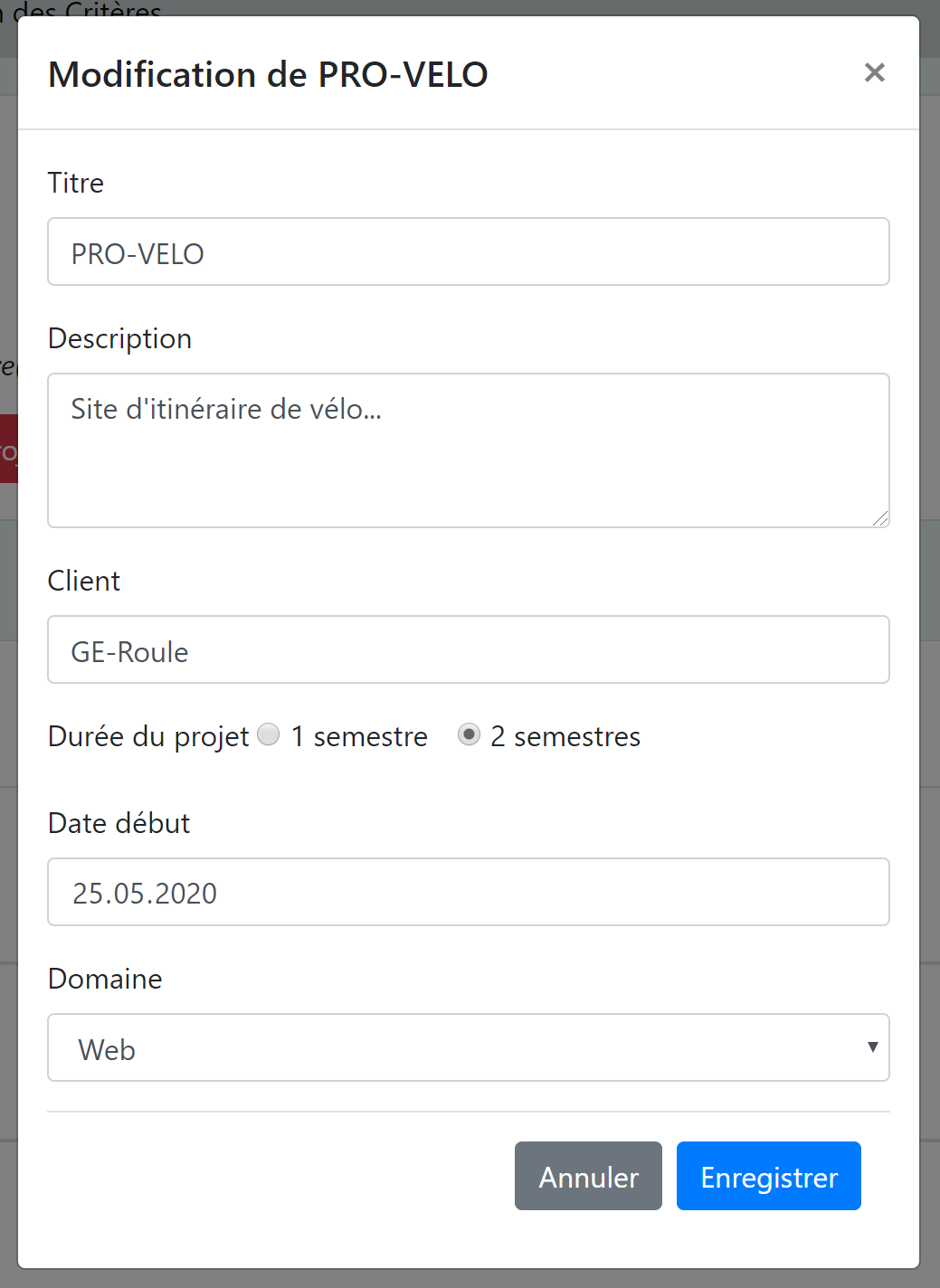


Figure 12 Modal de modification d'un projet (pour enseignants)

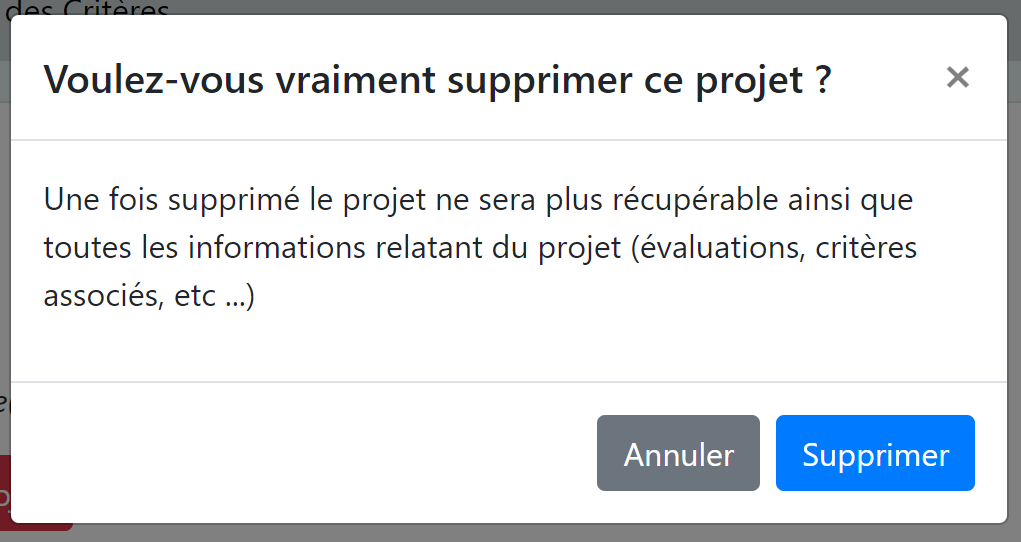


Figure 13 Modal pour la suppression de projet (pour enseignants)

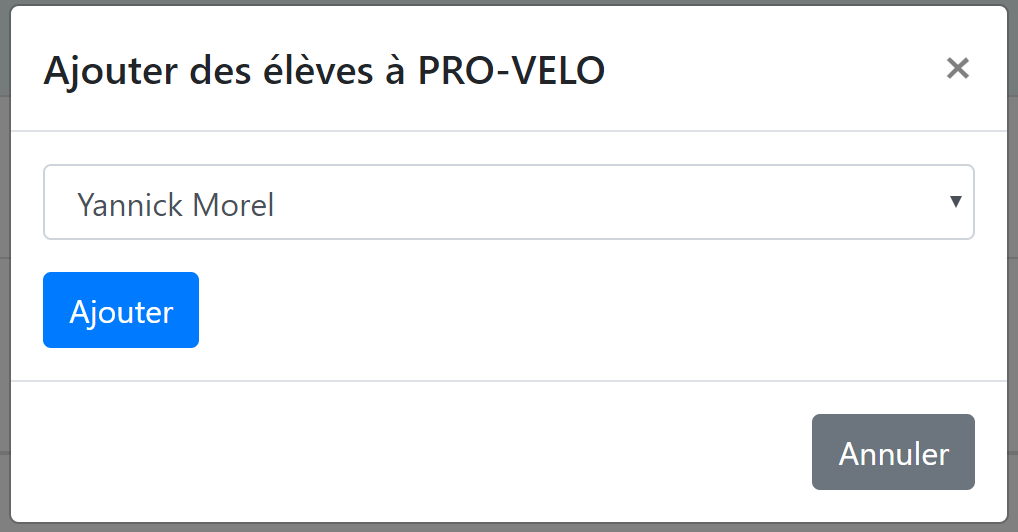


Figure 14 Modal d'ajout d'élèves dans un projet (pour enseignants)

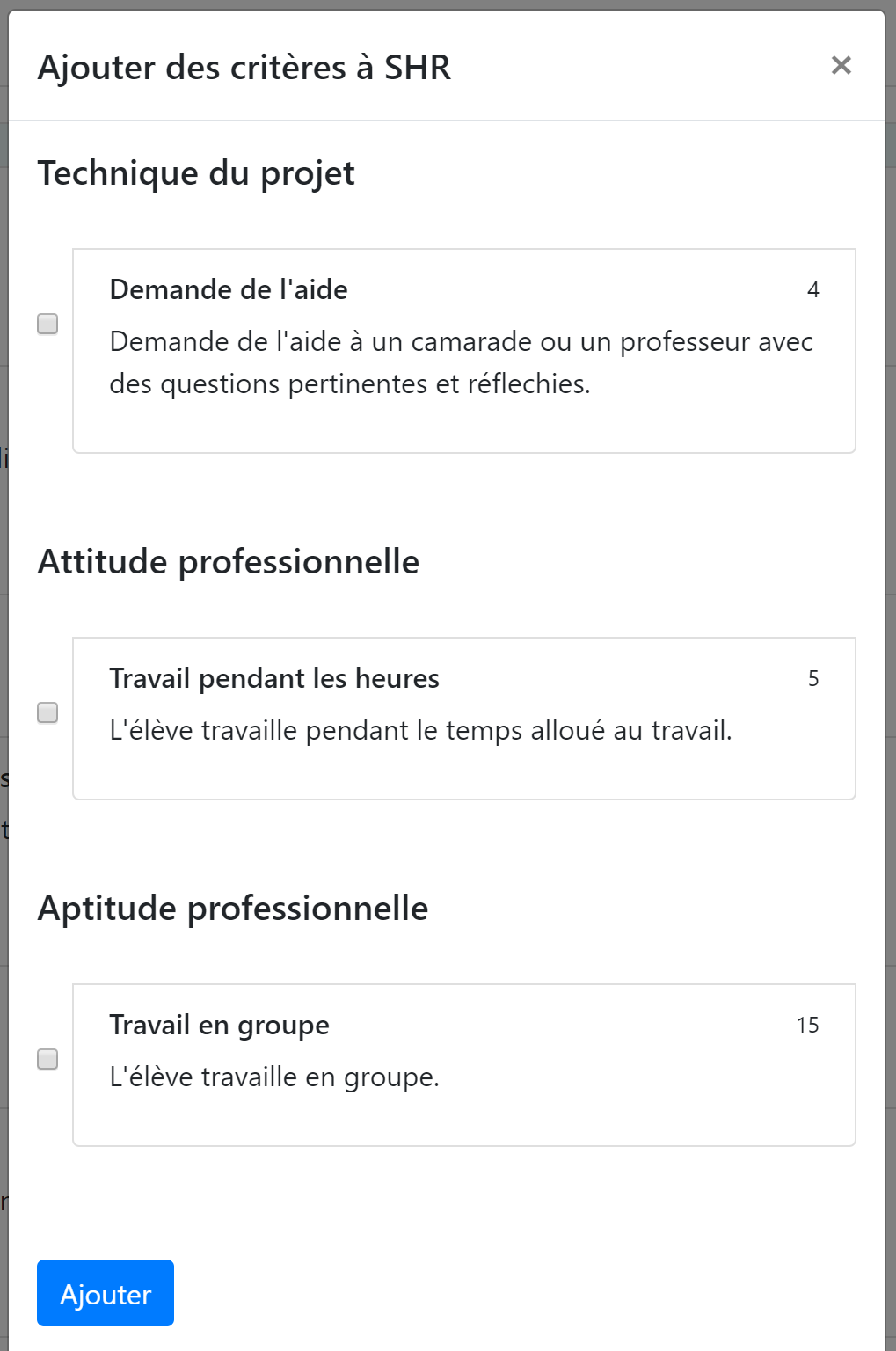


Figure 15 Modal d'ajout de critères dans un projet (pour enseignants)

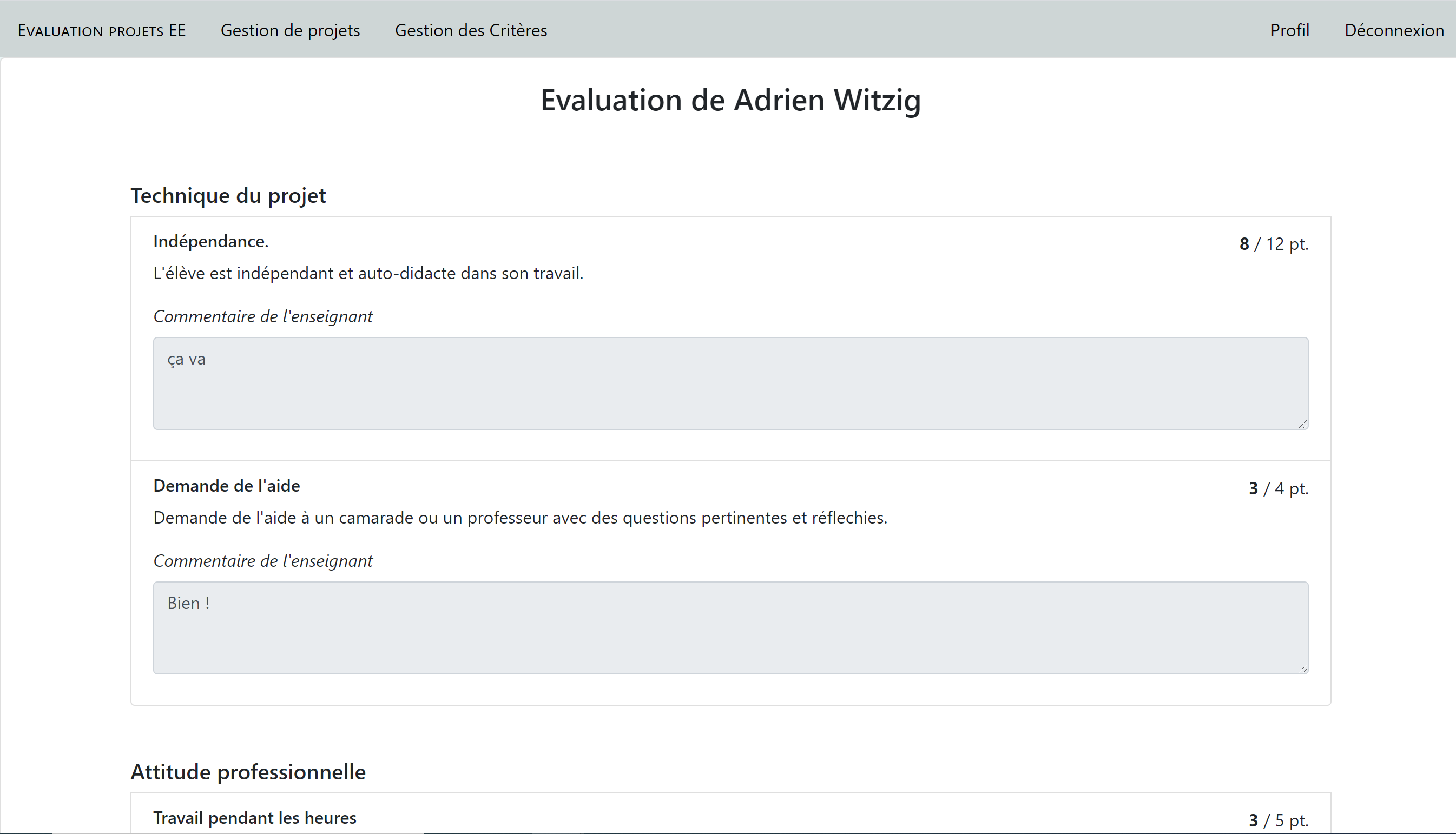


Figure 16 Page "voir évaluation" (pour enseignants) partie 1

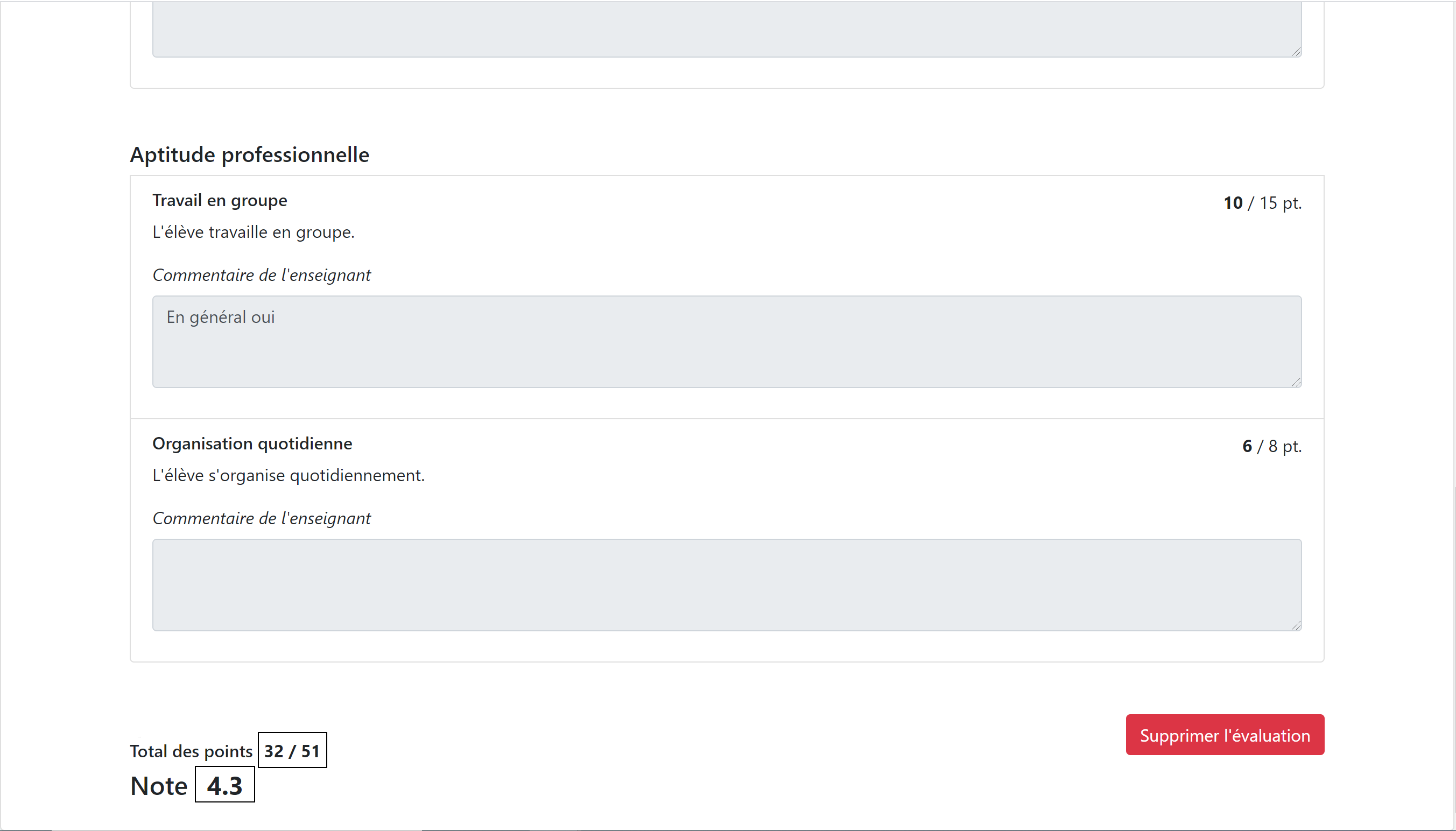


Figure 17 Page "voir évaluation" (pour enseignants) partie 2

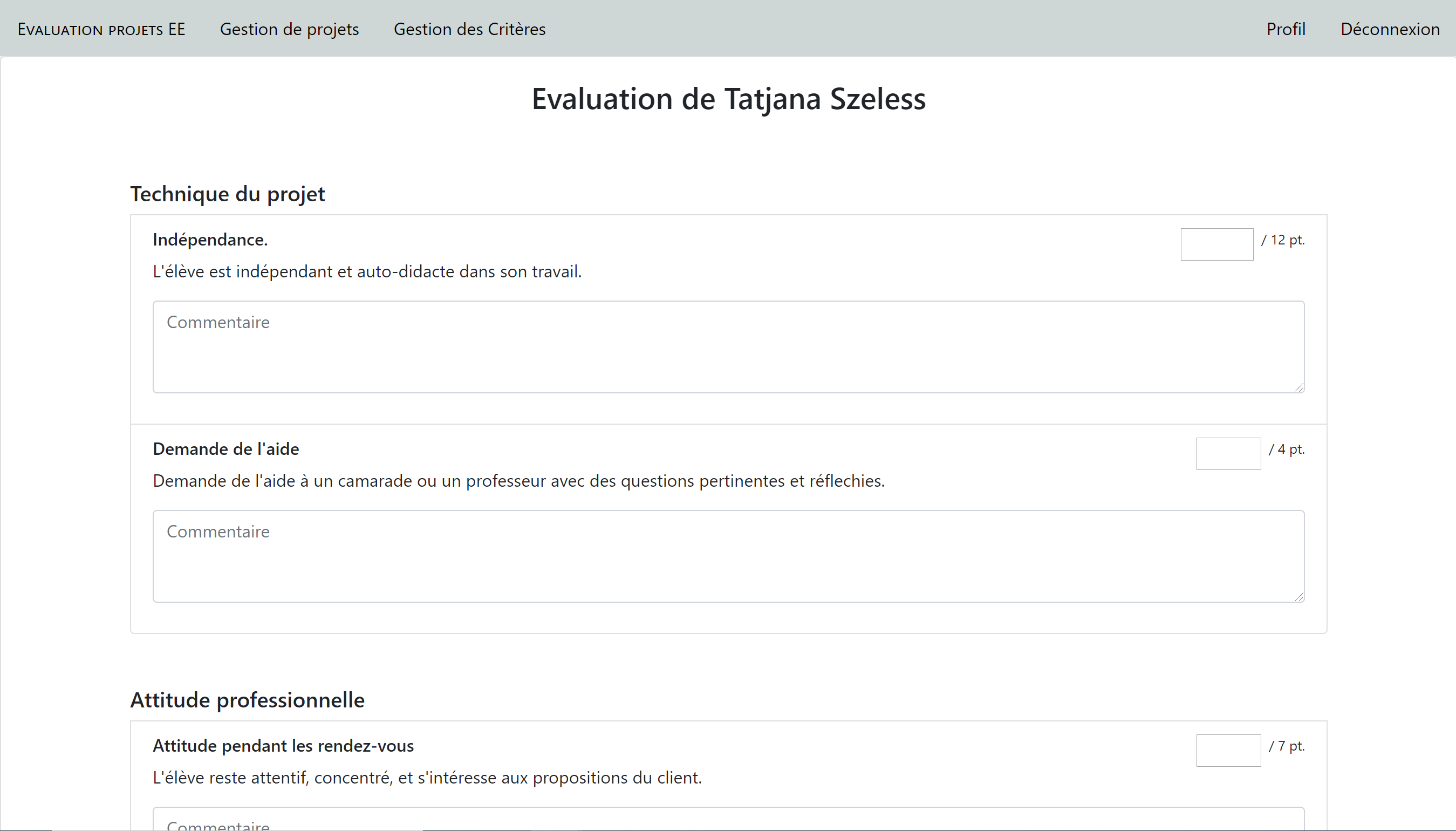


Figure 18 Page d'évaluation d'un élève (pour enseignants)

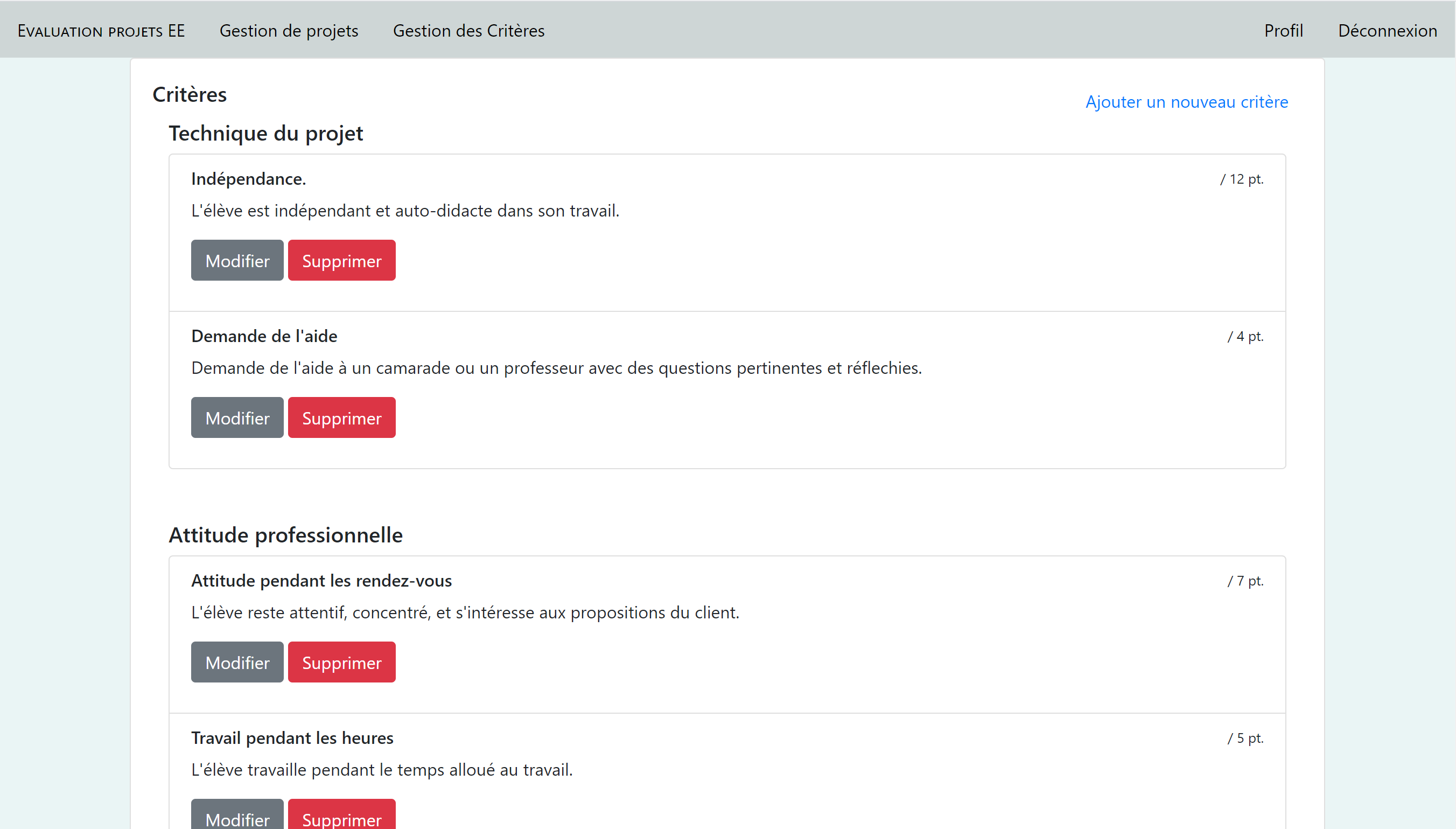


Figure 19 Page de gestion des critères (pour enseignants et administrateurs)

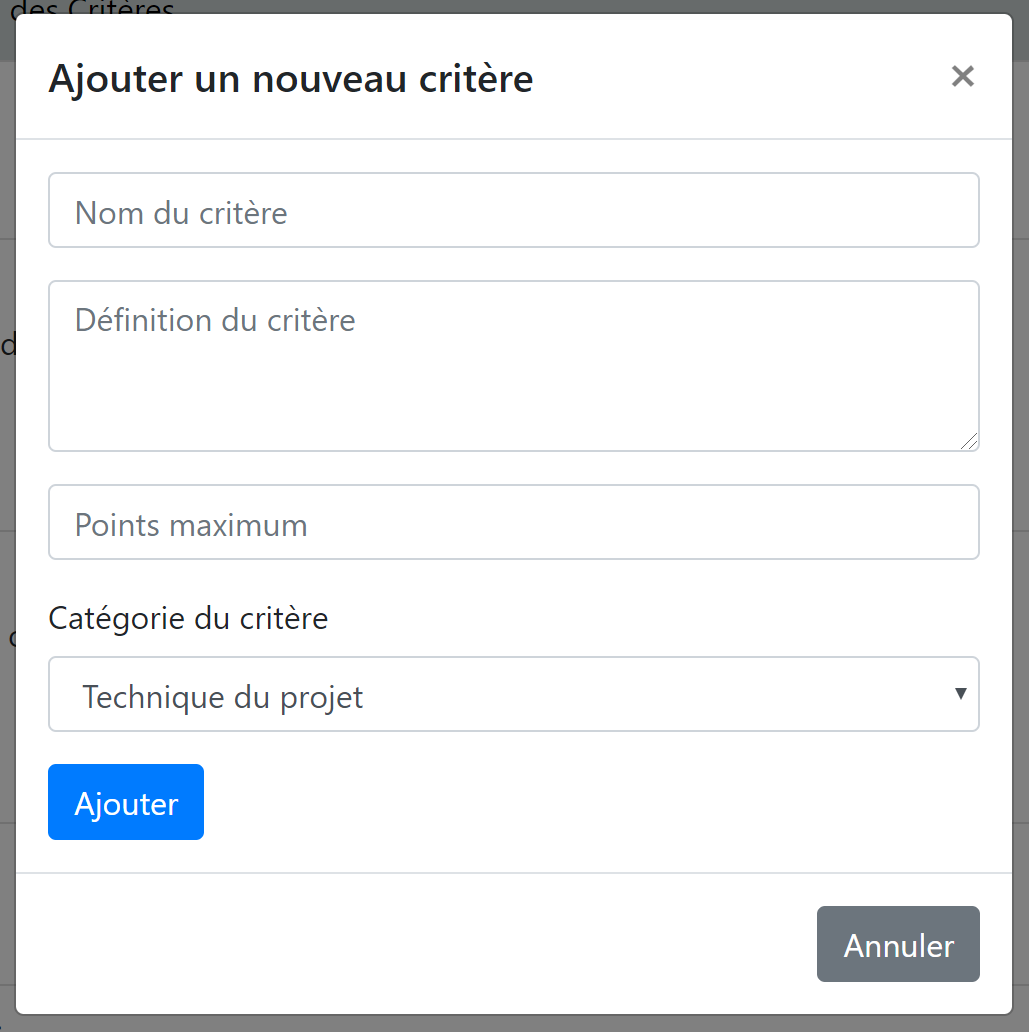


Figure 20 Modal d'ajout d'un nouveau critère



Figure 21 Page profil

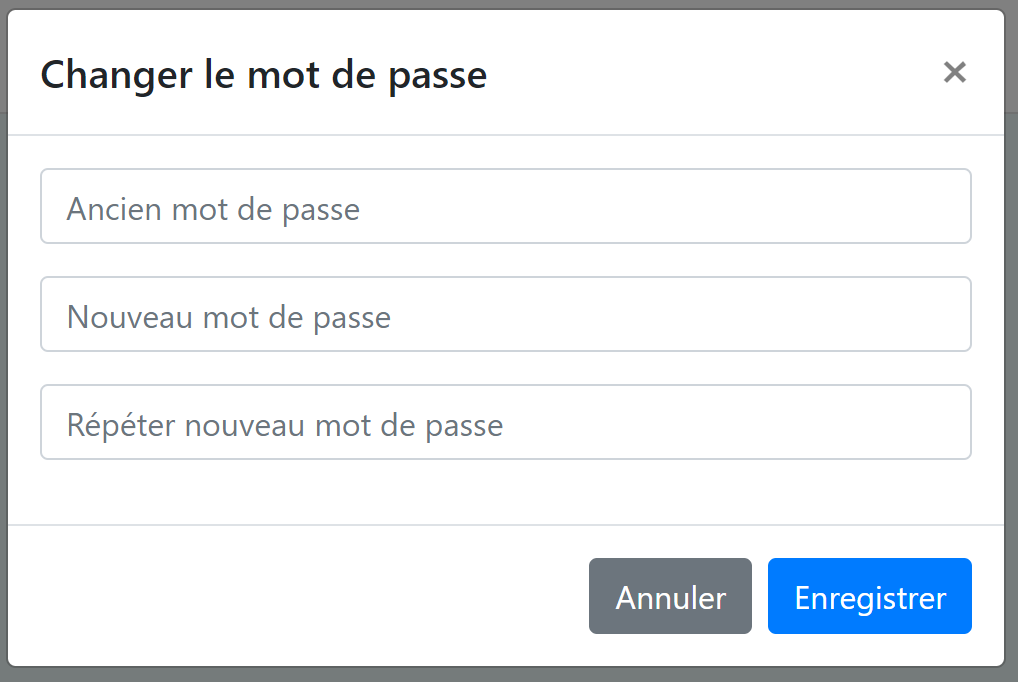


Figure 22 Modal de changement de mot de passe

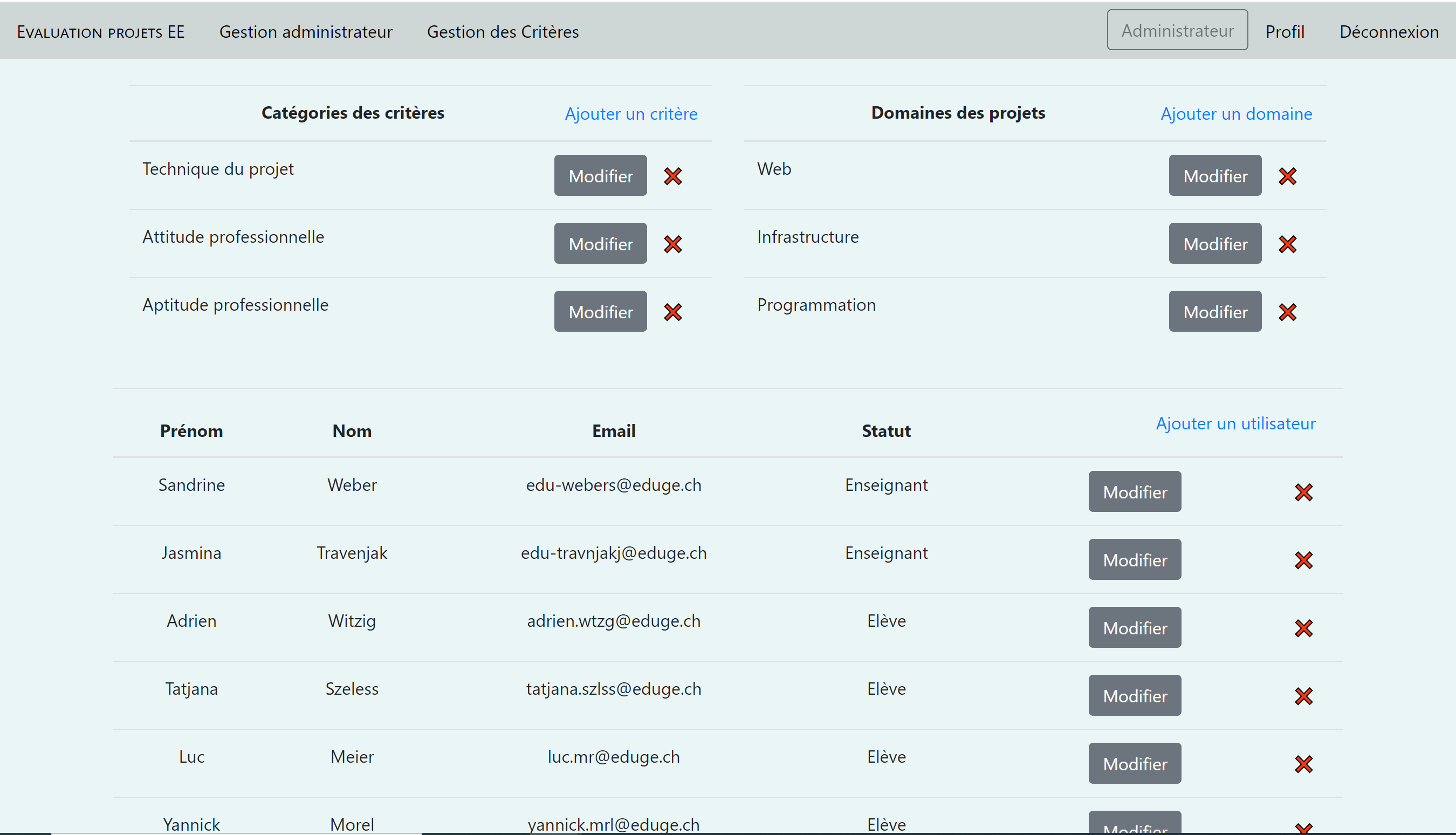


Figure 23 Gestion administrateur

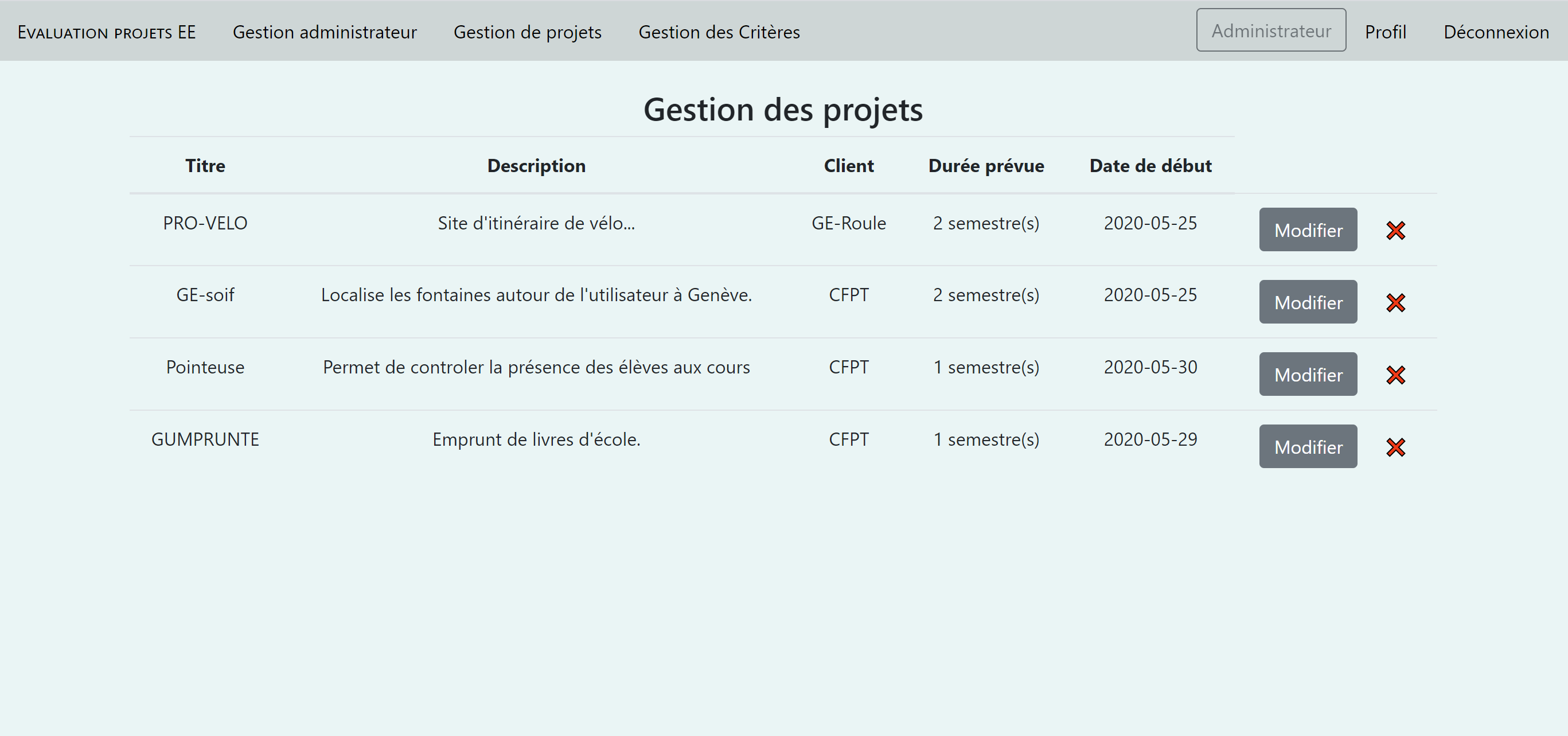


Figure 24 Gestion des projets (administrateur)

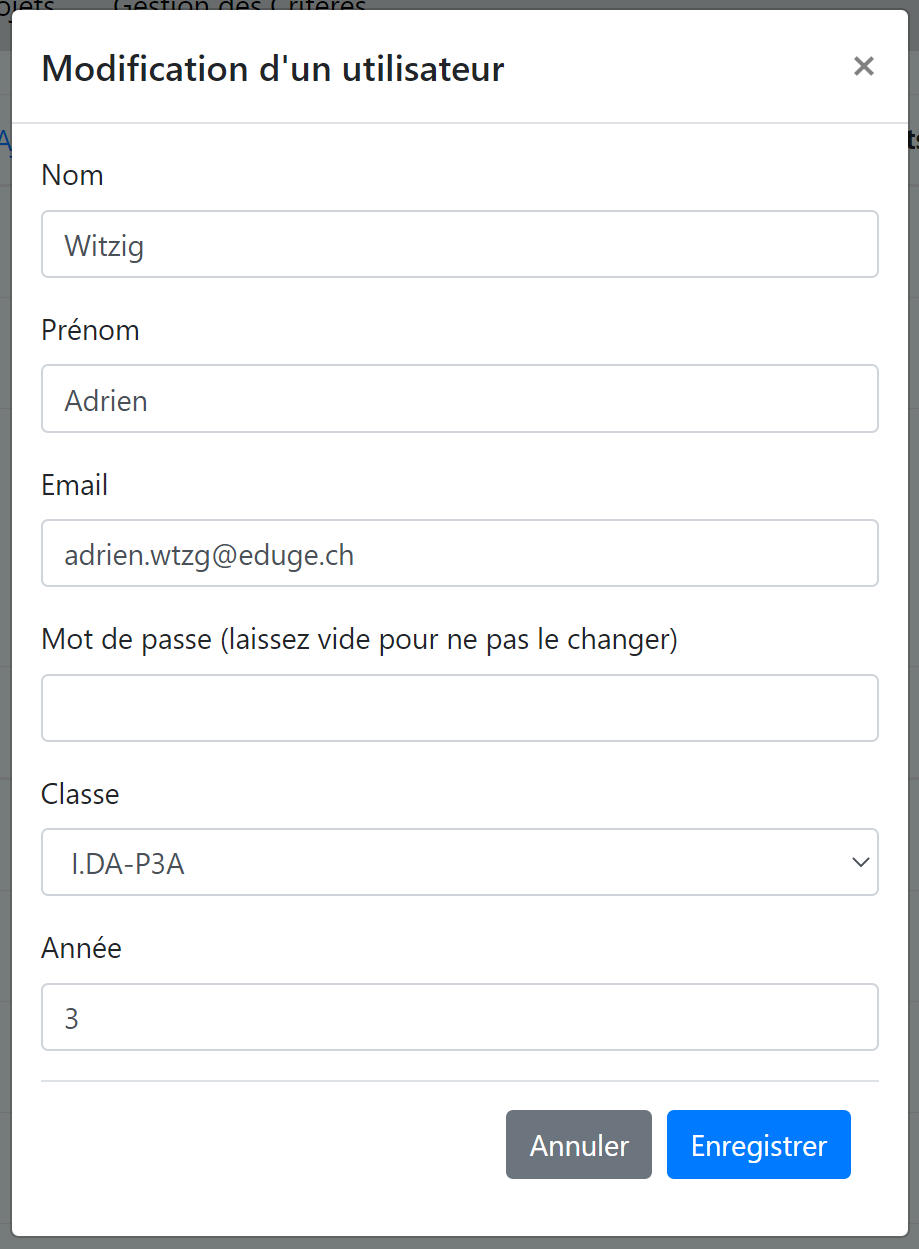


Figure 25 Modification d'un utilisateur

# Analyse organique

## Architecture du code

Ce site est conçu en MVC (Modèle Vue Contrôleur). Ce modèle suivant sépare la conception « front-end » (Vues) des conceptions « back-end » (Modèle, Contrôleur).

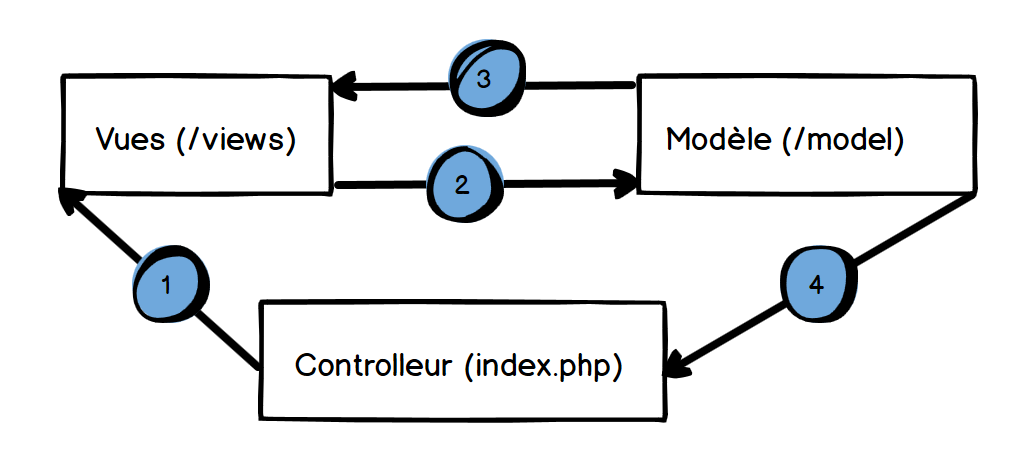


Figure 26 Architecture du code du site

1. Le **contrôleur** dirige vers une ou plusieurs **vues** du site.
2. Les **vues** demandent les informations au **modèle**.
3. Le **modèle** renvoie les informations à afficher aux **vues**.
4. Le **modèle** redirige l’utilisateur via le **contrôleur**.

## Arborescence de fichiers

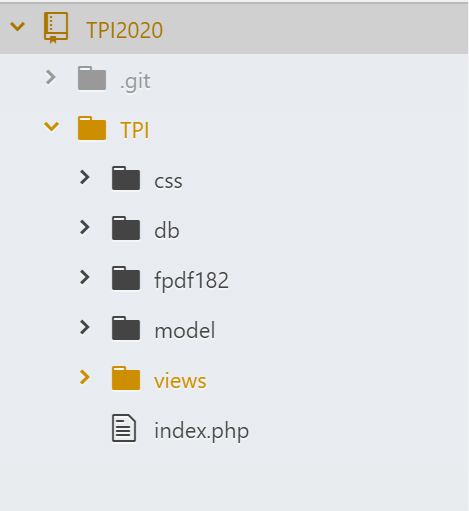


Figure 27 Arborescence de fichiers

TPI/

* **css/** : Dossier avec page de style .css
* **db/** : Contient un script de connexion à la base de données et la sauvegarde de la base de données
* **fpdf182/** : Fichiers de la libraire FPDF
* **model/** : Contient les scripts PHP
* **views/** : Contient les pages HTML
* **index.php** : Contrôleur PHP (fonction de routeur)

## Organigramme des pages

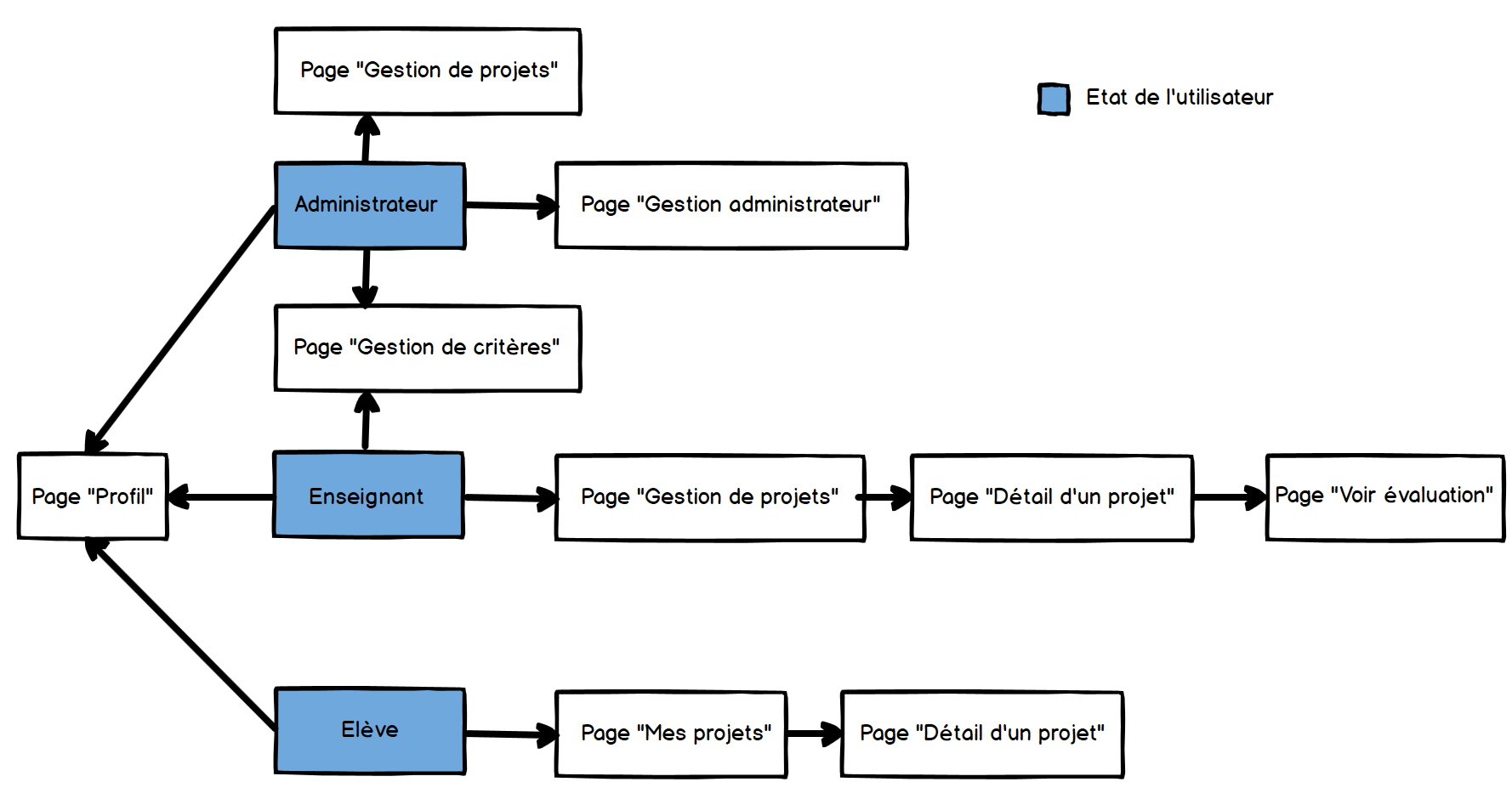


Figure 28 Organigramme des pages

## Implémentation de la sécurité entre les pages

Lorsque la connexion est réussie, l’utilisateur est redirigé sur des pages différentes selon le statut stocké dans la variable de session nommée statut.  


Figure 29 Segment du contrôleur (index.php) qui gère la redirection en fonction du statut

Dans le l’exemple (voir ci-dessus) suivant, le contrôleur (index.php) récupère la variable de session statut, crée lors de la connexion. De ce fait, j’utilise l’instruction switch(), qui permet d’exécuter certaines instructions différentes en fonction d’une variable, pour afficher certaines vues du site (fichiers PHP).

## Librairie FPDF

La libraire FPDF est un ensemble de fonctions PHP qui facilite la création de fichiers .PDF. Je l’ai téléchargé du site <http://www.fpdf.org/>.  
Après avoir initialiser la classe FPDF(), j’ai utilisé la méthode AddPage() qui ajoute une nouvelle page au document PDF. Ensuite je n’utilise plus que la fonction AddFont() qui permet de changer le style, la taille de la police et la méthode Cell() qui permet de crée une « cellule » qui seront les lignes du document.

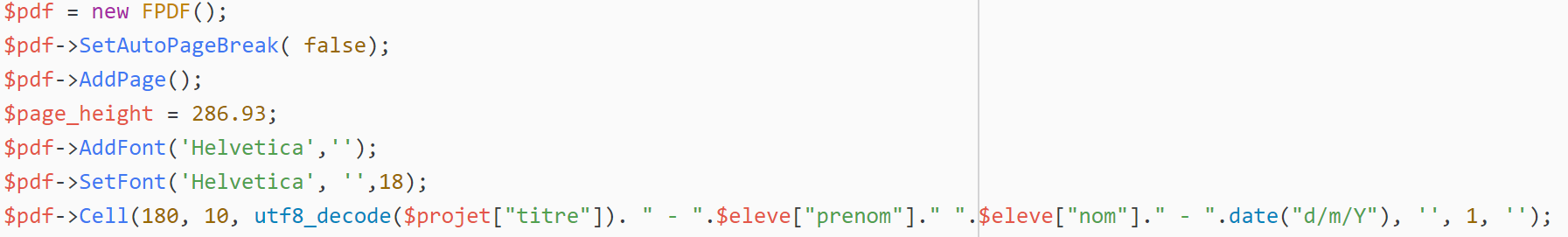


Figure 30 Segment de code pour l'impression de l'évaluation en format .PDF

Dans l’exemple ci-dessus, J’initialise la classe FPDF(), ensuite la fonction SetAutoPageBreak() ajoute une nouvelle page automatiquement lorsque c’est nécessaire. Pour finir j’ajoute comme « en-tête » du document, le **titre du projet**, le **prénom** et le **nom** de l’élève et la date du téléchargement.

## Librairie Bootstrap

La libraire Bootstrap est un ensemble de classes CSS qui permettent de styliser un site web beaucoup plus facilement. J’accède aux fichiers via un CDN[[4]](#footnote-4), qui récupèrent ceux-ci par le réseau. Je n’ai donc pas besoin de télécharger ces fichiers.  
Cette libraire est utile pour le style des pages donc les vues du site mais surtout pour rendre son site responsive[[5]](#footnote-5), ce qui est demandé dans le cahier des charges. Tous les éléments de Bootstrap sont responsive s’ils sont utilisés de la bonne manière.

Les principales fonctionnalités utilisées de Bootstrap sont listées ci-dessous.

La classe .container : Centre les éléments.

La classe .btn : Permet un style plus moderne du bouton (<button>[[6]](#footnote-6)).

La classe .card : Crée une section aux bords arrondis.

La classe .alert : Affiche un message à l’utilisateur avec des couleurs indicatives (selon le type de message à afficher).

La classe .list-group : Sert à afficher des éléments dans une liste.

Les classes .form-group et .form-control : Servent à afficher un style plus moderne aux éléments de la balise <form>, qui s’occupe des formulaires du site.

La classe .modal : Affiche une boite de dialogue, je l’ai beaucoup utilisée, car elle permet d’afficher, par exemple, un formulaire très interactif (voir par exemple, Figure 8 Modal d’ajout projets (pour enseignants)).

## JavaScript et libraire JQuery

J’utilise les langages JavaScript et JQuery dans mon projet. En effet, Je récupère les fichiers de JQuery via un CDN2 Voici quelques implémentations :

Dans l’exemple ci-dessus, J’ouvre une fenêtre modale (Bootstrap) depuis du code PHP. Lorsque l’information est envoyée du serveur au client, le code JQuery, est compilé et la modale s’affiche.

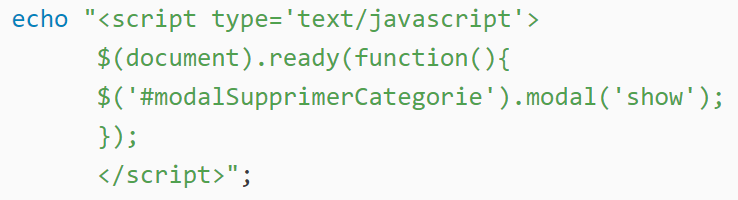


Figure 31 Lors d’une ouverture d’une fenêtre modal[[7]](#footnote-7) en PHP

Dans l’exemple ci-dessous, je n’affiche, les champs d’entrées utilisateur (balise <input>), classe et année que si l’utilisateur à modifier est élève.

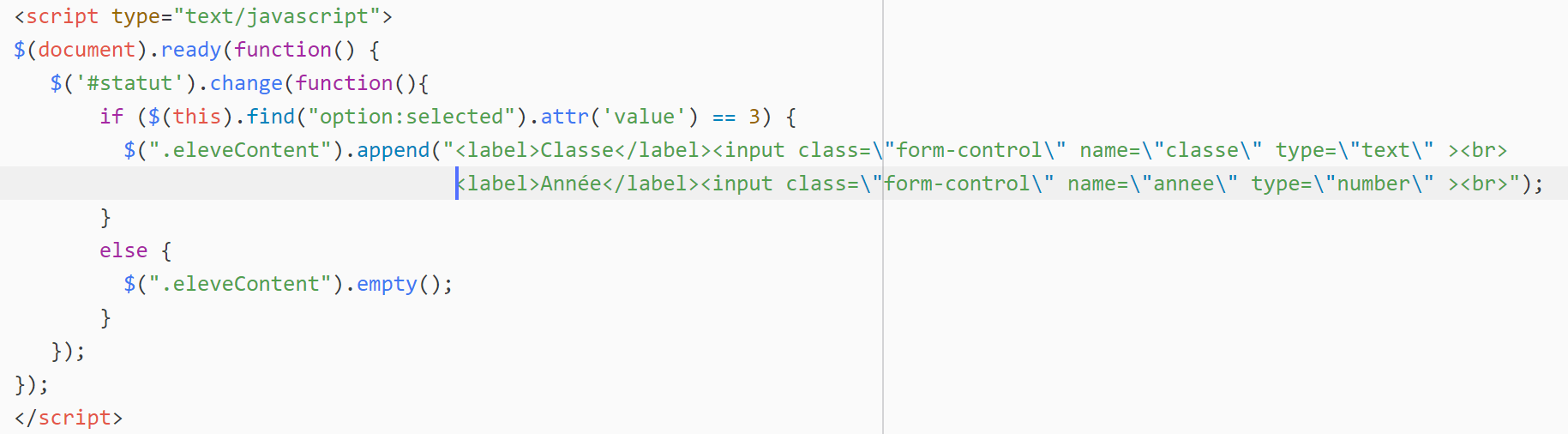


Figure 32 Lors d'ajout d'utilisateur, si l'administrateur ajoute un élève, les champs "classe" et "année" apparaissent

## Evaluation d’un élève

L’évaluation d’un élève se fait en additionnant le nombre de points obtenus et les points totaux sur chaque critère évalué. Ensuite j’applique la formule du barème fédéral, c’est-à-dire :

Cette note (sur 6) n’est pas stockée, car un enseignant ou un administrateur peut supprimer un critère d’évaluation (voir Figure 21 Page de gestion des critères ) même après l’évaluation d’un élève avec ce critère. Il faut savoir que la note sera mise à jour mais ne sera pas perturbée. En effet, l’évaluation de l’élève ne prendra pas en compte ce critère mais la note sera à nouveau calculée en fonction des autres critères.

## Fonctions et scripts PHP principaux

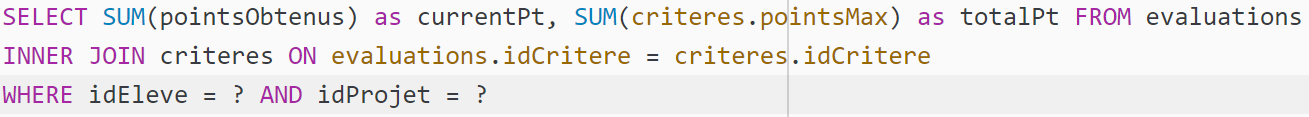
L’ensemble des fonctions PHP se trouvent dans le dossier /model de la racine.

### *connectDB()*

Cette fonction crée une connexion avec la base de données, elle permet d’effectuer par la suite des requêtes SQL dans celle-ci. Elle stocke donc les identifiants de connexion (voir 39).

### *getNote($idEleve, $idProjet)*

Cette fonction calcule la note d’un élève avec ces évaluations pour un projet. Le total des points obtenus/totaux est calculé via la requête suivante :



### *checkLogin.php*

Ce script vérifie les identifiants de connexions d’un utilisateur avec son **nom**, **prénom** et **mot de passe**. (voir 3.6 pour identifiants de connexion).

La requête récupère le mot de passe chiffré par l’algorithme sha1[[8]](#footnote-8) et le compare au mot de passe entré par l’utilisateur. Si les deux empreintes sont identiques la connexion est validée. Des variables de sessions[[9]](#footnote-9) PHP sont alors initialisées pour détecter les authentifications aux pages et droits d’accès.

### *estEvalue($idEleve, $idProjet)*

Cette fonction permet de savoir si un élève a été évalué dans un projet. Elle est essentielle car sinon le calcul de la note ne fonctionnerait pas. Si et seulement, si un élève a été évalué, l’on peut calculer sa note (voir 4.2.2 pour calcul de la note). La fonction retourne un booléen (= vrai ou faux).

### *getEleveToAdd($idProjet)*

Cette fonction permet de récupérer les élèves qui ne font pas encore parti du projet. Elle est appelée lorsqu’un enseignants veut ajouter un élève a son projet, alors elle ne retourne que les élèves qui n’en font pas partie.

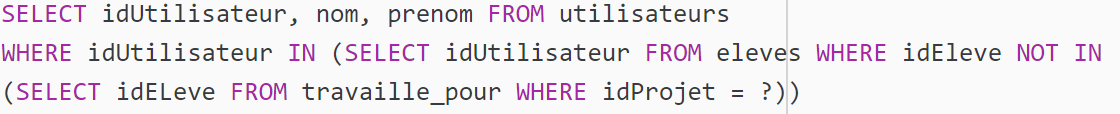


Figure 33 Requête d'affichage des élèves disponibles pour un projet

### *checkProjetExist($idProjet)*

Cette fonction permet de savoir si un projet existe avant d’en afficher son détail. En effet, il est possible que l’utilisateur arrive à changer la variable PHP $\_POST et dans ce cas, sans cette vérification, cela afficherait des erreurs. Elle retourne un booléen (= vrai ou faux).

### *evaluer.php*

Ce script permet aux enseignants de pouvoir évaluer des élèves dans un projet. Elle insère les informations d’évaluations dans la table evaluations et elle change le champ estEvalue dans la table travaille\_pour ce qui permet de savoir que l’élève a été évalué.

### *changeMotDePasse.php*

Ce script permet à l’utilisateur de changer son mot de passe. Elle vérifie l’ancien mot de passe et insère le nouveau mot de passe. La vérification est la même que pour le script *checkLogin.php* (voir 22).

### *logout.php*

Ce script déconnecte l’utilisateur. Etant donné que la connexion/vérification d’authenticité se fait par la session6 , il suffit donc de détruire la session comme ceci :

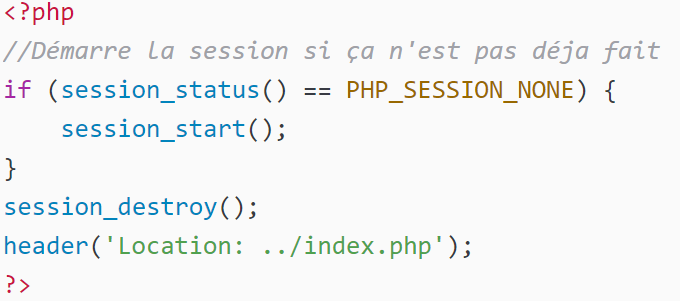


Figure 34 Script de déconnexion

### Scripts d’ajouts

Les scripts d’ajout sont nommées de la même manière. Voici un exemple : *addExemple.php.* Ces scripts fonctionnent tous de la même manière, ils ajoutent des données avec une requête INSERT INTO (mot clé MySQL désignant l’ajout de données) via un formulaire remplis par l’utilisateur. Voici une liste de ces scripts :

* *addUtilisateur.php (Ajout d’un utilisateur)*
* *addProject.php (Ajout d’un projet)*
* *addNewCritere.php (Ajout d’un nouveau critère)*
* *addEleveToProjet.php (Ajout d’un élève à un projet)*
* *addDomaine.php (Ajout d’un domaine de projet)*
* *addCritereToProjet.php (Ajout d’un critère à un projet)*

### Scripts de modification

Les scripts de modification sont nommés de la même manière. Voici un exemple : *updateExemple.php*. Ces scripts fonctionnent tous de la même manière, ils modifient les données avec une requête UPDATE (mot clé MySQL désignant la mise à jour de données) via un formulaire modifié par l’utilisateur. Voici une liste de ces scripts

* *updateUtilisateur.php (Modification d’un utilisateur)*
* *updateProjet.php (Modification d’un projet)*
* *updateCritere.php (Modification d’un critère)*
* *updateCategorie (Modification d’une catégorie de critère)*
* *updateDomaine.php (Modification d’un domaine de projet)*

### Scripts de suppression

Les scripts de suppression sont nommés de la même manière. Voici un exemple : *deleteExemple.php*. Ces scripts fonctionnent tous de la même manière, ils suppriment les données avec une requête DELETE FROM (mot clé MySQL désignant la suppression de données) via un formulaire de confirmation par l’utilisateur (voir 7 pour méthodes de suppression). Voici une liste de ces scripts

* *deleteUtilisateur.php (Suppression d’un utilisateur)*
* *deleteProjet.php (Suppression d’un projet)*
* *deleteCritere.php (Suppression d’un critère)*
* *deleteCritereProjet.php (Suppression d’un critère d’un projet)*
* *deleteEleveProjet.php (Suppression d’un élève d’un projet)*
* *deleteCategorie (Suppression d’une catégorie de critère)*
* *deleteDomaine.php (Suppression d’un domaine de projet)*
* *deleteEvaluation.php (Suppression d’une évaluation)*

## Modèle logique de données

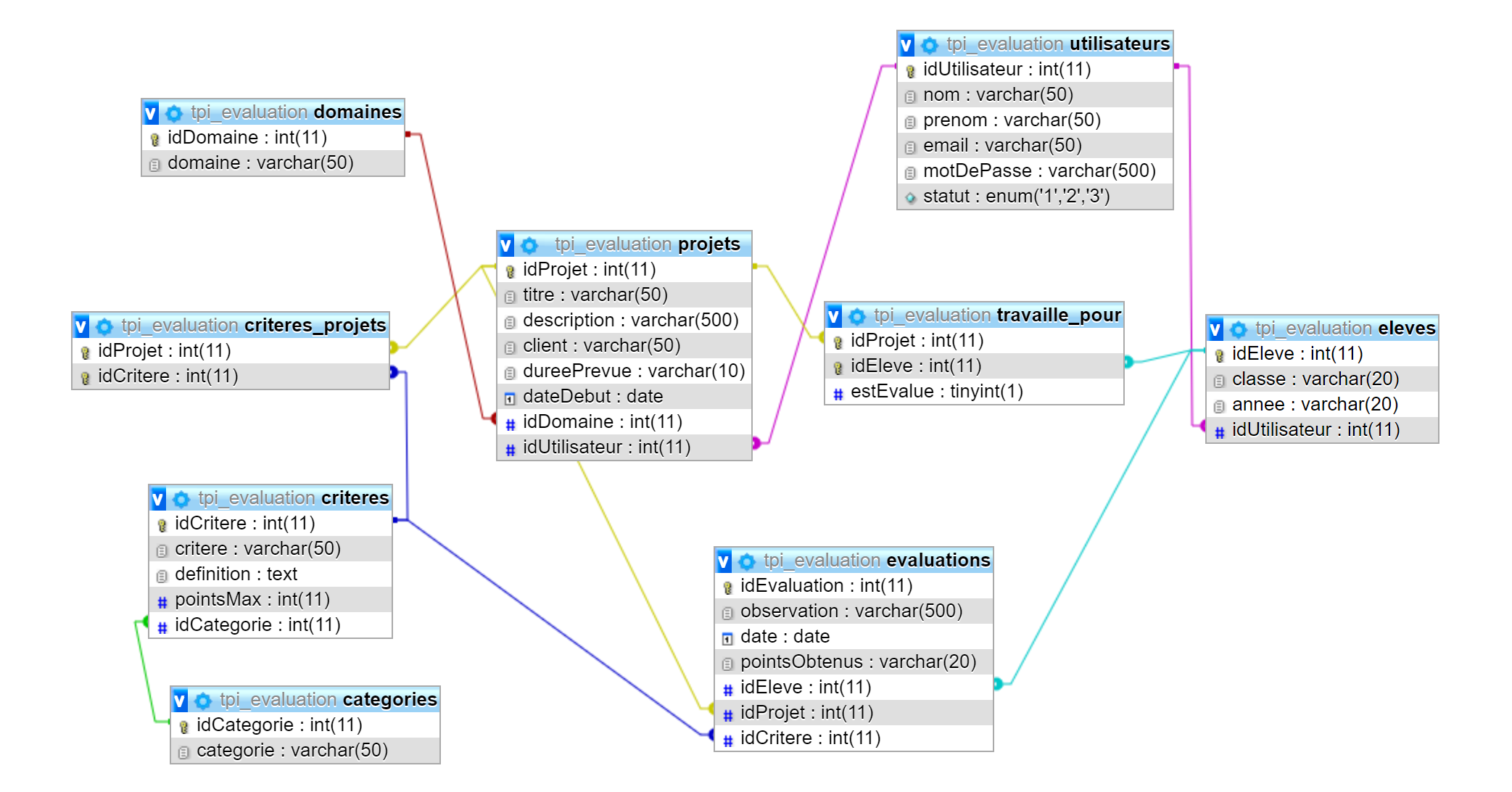


Figure Modèle logique de données

Pour la traduction du MCD (Modèle conceptuel de données, voir 4) en MLD (Modèle logique de données), il a fallu rajouter la table travaille\_pour, car un projet peut avoir plusieurs élèves et un élève peut faire partie de plusieurs projets. Il a fallu également rajouter la table criteres\_projets car un critère peut être utilisé par plusieurs projets et un projet peut utiliser plusieurs critères.

## Suppressions de données

Lorsque des données sont supprimées, il faut supprimer leurs occurrences dans d’autres tables de la base ou alors empêcher la suppression à cause des répercutions que cela peut avoir. L’utilisateur n’est souvent pas au courant de cela, c’est pourquoi avant chaque suppression j’affiche une modal de confirmation où j’explique les restrictions et conséquences de la suppression.

Voici une liste qui répertorie les suppressions qui ont lieu lors d’une action.

Lors de la suppression d’une catégorie de projet :

* **Si** la catégorie ne possède pas de critères **alors** elle est supprimée.

Lors de la suppression d’un domaine de projet :

* **Si** le domaine n’appartient à aucuns projets **alors** il est supprimé.

Lors de la suppression d’un utilisateur :

L’administrateur ne peut pas supprimer un autre administrateur mais il peut supprimer les élèves ou les enseignants (voir ci-dessous).

Lors de la suppression d’un élève :

* On doit supprimer d’abord ses occurrences dans la table travaille\_pour et evaluations.
* Ensuite on peut supprimer dans la table eleves.
* Pour finir, on supprime dans la table utilisateurs.

Lors de la suppression d’un enseignant :

* **Si** l’enseignant ne possède pas de projets **alors** il est supprimé

## Tables de données

### Table UTILISATEURS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nom** | **Type** | **Null** | **Défaut** | **Associé à** |
| idUtilisateur | int(11) | Non |  |  |
| nom | varchar(50) | Non |  |  |
| prenom | varchar(50) | Non |  |  |
| email | varchar(50) | Non |  |  |
| motDePasse | varchar(500) | Non |  |  |
| statut | enum(‘1’,’2’,’3’) | Non |  |  |

La table utilisateurs contient toutes les informations relatives aux utilisateurs, c’est-à-dire, les **administrateurs,** les **enseignants** et les **élèves.** Elle stocke son **nom,** son **prénom**, son **email,** son **mot de passe** et le **statut** de connexion.

### Table ELEVES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nom** | **Type** | **Null** | **Défaut** | **Associé à** |
| idEleve | int(11) | Non |  |  |
| classe | varchar(20) | Non |  |  |
| annee | varchar(20) | Non |  |  |
| idUtilisateur | Int(11) | Non |  | utilisateurs  (idUtilisateur) |

La table eleves contient les informations complémentaires d’un utilisateur élève. C’est-à-dire, sa **classe** et son **année** de formation. Chaque élève est associé à un utilisateur par l’identifiant idUtilisateur.

### Table TRAVAILLE\_POUR

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nom** | **Type** | **Null** | **Défaut** | **Associé à** |
| IdProjet | int(11) | Non |  | projets  (idProjet) |
| idEleve | int(11) | Non |  | eleves  (idEleve) |
| estEvalue | bool | Non | 0 |  |

La table travaille\_pour contient les informations de liaisons entre les élèves et les projets. En effet, lorsqu’un enseignant ajoute un élève à un projet, cette information est stockée dans cette table avec les identifiants idProjet et idEleve. Elle contient également, un champ estEvalue booléen (= vrai ou faux) qui indique si l’élève a été évalué dans un projet.

### Table PROJETS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nom** | **Type** | **Null** | **Défaut** | **Associé à** |
| idProjet | int(11) | Non |  |  |
| titre | varchar(50) | Non |  |  |
| description | varchar(500) | Non |  |  |
| client | varchar(50) | Non |  |  |
| dureePrevue | varchar(10) | Non |  |  |
| dateDebut | dateTime | Non |  |  |
| idDomaine | Int(11) | Non |  | domaines  (idDomaine) |
| idUtilisateur | Int(11) | Non |  | Utilisateurs  (idUtilisateur) |

La table projet inclus les informations relatives aux projets ajoutés par un enseignant. De ce fait, elle contient le **titre** du projet, la **description** du projet, le nom du **client**, la **durée prévue**, la **date de début** du projet. Elle est reliée à la table domaines par l’identifiant idDomaine et à la table utilisateurs avec l’identifiant idUtilisateur qui indique l’enseignant qui a créé le projet.

### Table DOMAINES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nom** | **Type** | **Null** | **Défaut** | **Associé à** |
| idDomaine | int(11) | Non |  |  |
| domaine | varchar(50) | Non |  |  |

La table domaines stocke le nom des domaines de projets. En effet, les projets ont des domaines particuliers (*exemple : Web*).

### Table CRITERES\_PROJETS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nom** | **Type** | **Null** | **Défaut** | **Associé à** |
| idProjet | int(11) | Non |  | projets  (idProjet) |
| idCritere | int(11) | Non |  | criteres  (idCritere) |

La table criteres\_projets renferme les informations de liaisons des tables projets et criteres. Dès qu’un critère est ajouté à un projet, il est stocké dans cette table avec les identifiants idProjet et idCritere.

### Table CRITERES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nom** | **Type** | **Null** | **Défaut** | **Associé à** |
| idCritere | int(11) | Non |  |  |
| critere | varchar(50) | Non |  |  |
| definition | text | Non |  |  |
| pointsMax | int(11) | Non |  |  |
| idCategorie | int(11) | Non |  | categorie  (idCategorie) |

La table criteres contient les informations des critères d’évaluations. En effet, on y trouve le **nom** du critère, la **définition** du critère, le nombre de **points maximum**, et l’identifiant idCategorie qui représente la catégorie du critère d’évaluation.

### Table CATEGORIES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nom** | **Type** | **Null** | **Défaut** | **Associé à** |
| idCategorie | int(11) | Non |  |  |
| categorie | varchar(50) | Non |  |  |

La table categories comporte le **nom**  de la catégorie d’un critère d’évaluation. Lorsqu’un critère d’évaluation est créé, l’enseignant choisit à quelle catégorie il appartient.

### Table EVALUATIONS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nom** | **Type** | **Null** | **Défaut** | **Associé à** |
| idEvaluation | int(11) | Non |  |  |
| observation | varchar(500) | Non |  |  |
| date | text | Non |  |  |
| pointsObtenus | varchar(20) | Non |  |  |
| idEleve | int(11) | Non |  | categories  (idCategorie) |
| IdProjet | Int(11) | Non |  | projets  (idProjet) |
| IdCritere | Int(11) | Non |  | criteres  (idCritere) |

La table evaluations retient les évaluations, par critère d’évaluation, des élèves dans un projet. Effectivement, on y trouve l’**observation** du critère, la **date** d’évaluation, les **points obtenus**. En outre, elle est reliée aux tables eleves, projets et criteres, par les identifiants idEleve, idProjet et idCritere.

# Tests

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Utilisateur non authentifié | N° | Date | Résultat attendu | Résultat |
| 1 | 05.06.2020 | Sur la page de connexion, appuyer sur « Se connecter » sans renseigner un des champs (Nom, Prénom, Mot de passe), une erreur s’affiche et demande de renseigner le champ. | OK |
| 2 | 05.06.2020 | Renseigner un Nom et Prénom et Mot de passe qui n’existent pas et un vrai Nom et Prénom mais un mauvais Mot de passe et appuyer sur « Se connecter ». Dans les deux cas, une erreur « Les identifiants de connexion sont incorrects » doit s’afficher. | OK |
| 3 | 05.06.2020 | Lorsque l’utilisateur non-authentifié essaie d’accéder à une page via l’url en remplaçant la variable $\_GET 🡪 page. Dans tous les cas, l’utilisateur doit être redirigé sur la page de login. | OK |
| Enseignant | 4 | 05.06.2020 | Dans la page « gestion de projets » les projets s’affichent correctement et lors d’un ajout de projet tous les champs sont requis. | OK |
| 5 | 05.06.2020 | Après le N°4, on peut cliquer sur « Aller à » et on arrive sur le détail du projet. | OK |
| 6 | 05.06.2020 | Sur la page « détail d’un projet », nous ne pouvons pas modifier le projet et ne pas renseigner un champ. | OK |
| 7 | 05.06.2020 | On peut supprimer le projet et cela supprime toutes les informations relatant du projet (évaluations, associations d’élèves, associations de projets | OK |
| 8 | 05.06.2020 | Dans le détail du projet, on peut ajouter seulement les élèves qui ne font pas partie du projet. | OK |
| 9 | 05.06.2020 | Dans le détail du projet, on peut enlever un élève du projet s’il n’a pas encore été évalué. | OK |
| 10 | 05.06.2020 | Dans le détail du projet, si aucun élève n’a été évalué, on peut ajouter plusieurs critères existants qui ne font pas partie du projet. | OK |
| Enseignant | 11 | 05.06.2020 | Dans le détail du projet, si aucun élève n’a été évalué, on peut enlever les critères des projets | OK |
| 12 | 05.06.2020 | Dans le détail du projet, si un élève a été évalué on peut voir le bouton « voir évaluation » et voir sa note. | OK |
| 13 | 05.06.2020 | Si on appuie sur « voir évaluation », on peut voir l’évaluation de l’élève, le total des points et télécharger son évaluation | OK |
| 14 | 08.06.2020 | Dans le détail du projet, si un élève n’a pas été évalué, nous pouvons l’évaluer. Lors de l’évaluation nous ne pouvons pas mettre plus de points que le nombre de points maximum du critère | OK |
| 15 | 08.06.2020 | Dans la page gestion de critères, On peut voir les critères existants, en ajouter/modifier(en renseignant les champs !) et les supprimer (La suppression se fait en cascade) | OK |
| 16 | 08.06.2020 | Dans la page « profil », l’enseignant peut modifier son mot de passe et cela affiche une erreur si l’ancien mot de passe n’est pas juste ou que le nouveau mot de passe et la répétition sont différents. | OK |
| 17 | 08.06.2020 | Si l’enseignant appuie sur le lien « Déconnexion », l’utilisateur est déconnecté et il ne peut plus accéder aux pages même en modifiant l’URL, | OK |
| Elève | 18 | 08.06.2020 | L’élève peut se connecter avec son nom, prénom et mot de passe. | OK |
| 19 | 08.06.2020 | L’élève ne peut pas accéder à une page pour enseignants ou admin. | OK |
| 20 | 08.06.2020 | L’élève peut consulter les projets dans lesquels il a été ajouté | OK |
| 21 | 08.06.2020 | Si l’élève a été évalué, il peut consulter son évaluation et la télécharger. | OK |
| 22 | 08.06.2020 | Si l’élève n’a pas été évalué, un message s’affiche | OK |
| 23 | 08.06.2020 | L’élève peut voir ses collaborateurs dans le détail du projet. | OK |
| 24 | 08.06.2020 | Lorsque l’élève télécharge son évaluation, le fichier .PDF s’adapte aux nombres de critères et ne bug pas | OK |
| Elève | 25 | 08.06.2020 | Après le N°24, l’évaluation en .PDF affiche correctement les accents. | OK |
| 26 | 08.06.2020 | Dans le profil, l’élève peut voir sa classe | OK |
| 27 | 08.06.2020 | Dans la page « profil », l’élève peut modifier son mot de passe et cela affiche une erreur si l’ancien mot de passe n’est pas juste ou que le nouveau mot de passe et la répétition sont différents. | OK |
| Administrateur | 28 | 08.06.2020 | L’administrateur peut se connecter avec son nom, prénom et mot de passe |  |
| 29 | 08.06.2020 | Dans la page gestion administrateur, l’administrateur ne peut pas ajouter des catégories de critères ou domaines de projets sans nom | OK |
| 30 | 08.06.2020 | Dans la page gestion administrateur, l’administrateur ne peut pas ajouter un utilisateur sans nom ou autre champs requis (tous) | OK |
| 31 | 08.06.2020 | Dans la page gestion administrateur, l’administrateur ne peut pas supprimer une catégorie de critère si des critères en font partie. |  |
| 32 | 08.06.2020 | Dans la page gestion administrateur, l’administrateur ne peut pas supprimer une catégorie de critère si des critères en font partie. | OK |
| 33 | 08.06.2020 | Dans la page gestion administrateur, l’administrateur ne peut pas supprimer un domaine de projet s’il est le domaine d’un projet en cours. | OK |
| 34 | 08.06.2020 | Dans la page gestion d’administrateur, l’administrateur ne peut pas supprimer un utilisateur si l’utilisateur est un enseignant et qu’il possède un projet. | OK |
| 35 | 08.06.2020 | Dans la page gestion d’administrateur, l’administrateur ne peut pas ajouter deux utilisateurs avec le même prénom et nom | OK |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Administrateur | 36 | 08.06.2020 | Dans la page gestion de projets, l’administrateur peut supprimer un projet et cela supprime toutes ses occurrences dans la base de données. | OK |
| 37 | 08.06.2020 | Dans la page gestion de projets, l’administrateur peut modifier un projet et il peut directement observer la modification | OK |
| 38 | 08.06.2020 | Dans la page gestion de critères, l’administrateur peut supprimer un critère et toutes les occurrences de ce critère sont supprimées dans la base de données. | OK |

Tableau 1 Tests

# Conclusion

## Installation du projet

Copiez le dossier /TPI et déposez le dans votre serveur apache (EasyPhp ou XAMP, par exemple).

Exécutez le script SQL tpi\_evaluation.sql ou importer celui-ci dans PhpMyAdmin.

Les informations de connexions à la base de données se trouvent dans le fichier **/TPI/db/databaseConnection.php**, vous n’avez rien à changer dans ce fichier, mais les identifiants de connexions à la base de données sont les suivants :

* Nom d’utilisateur : tpi2020
* Mot de passe : Super

Données présentes dans la base de données :

Administrateur :

|  |  |  |
| --- | --- | --- |
| Prénom | Nom | Mot de passe |
| John | Doe | super |

Tableau 2 Données administrateur

Enseignants :

|  |  |  |
| --- | --- | --- |
| Prénom | Nom | Mot de passe |
| Sandrine | Weber | super |
| Jasmina | Travenjak | super |

Tableau 3 Données enseignants

Élèves :

|  |  |  |
| --- | --- | --- |
| Prénom | Nom | Mot de passe |
| Adrien | Witzig | super |
| Tatjana | Szeless | super |
| Luc | Meier | super |
| Yannick | Morel | super |

Tableau 4 Données élèves

## Problèmes rencontrés

Tout au long de mon travail, j’ai rencontrés divers problèmes qui m’ont freiné dans l’avancement de mon projet.   
Heureusement, j’ai su garder mon calme face à ceux-ci et m’organiser dans leur résolution.   
Si un problème persiste, Je m’aide des forums (comme <http://stackoverflow.com>) ou de documentation des langages (comme <http://php.net> pour PHP).   
Si cette aide ne me suffit pas et que le problème persiste encore, je cherche un moyen d’arriver au même résultat différemment, sans perte de robustesse[[10]](#footnote-10) dans mon application.

Description des différents problèmes rencontrés :

J’ai eu des bugs concernant la classe .modal de Bootstrap. La résolution était fastidieuse car je ne comprenais pas pourquoi la fenêtre ne s’ouvrait pas. Il s’agissait en fait d’une balise HTML ouverte dans un autre fichier et que je ne refermais pas.  
De plus, dans fichier PDF que je génère pour des enseignants ou des élèves, j’avais de la peine à gérer les accents. J’ai fait des recherche sur le forum <http://stackoverflow.com>, et j’ai découvert une instruction PHP utf8\_decode(). Celle-ci permet de convertir une chaine de caractère UTF-8 en ISO-8859-1[[11]](#footnote-11)  
J’ai également eu de la peine pour l’ajout d’un utilisateur. En effet, Lorsque l’administrateur choisit s’il veut ajouter un élève ou un enseignant et qu’il choisit d’ajouter un élève, les champs classe et année doivent être ajoutés au formulaire.  
Après mûre réflexion, j’ai trouvé ça plus dynamique de le faire en JavaScript pour éviter le rafraichissement de la page (voir 5.7).

## Améliorations possibles

Ce projet a encore beaucoup de potentiel et ce que j’ai réalisé en 11 jours, peut être perfectionné et amélioré. J’aurais aimé implémenter les idées citées ci-dessous, mais j’ai préféré me concentrer sur les critères demandés dans le cahier des charges. J’espère qu’un ou plusieurs autres élèves pourront reprendre le projet pour l’améliorer ou l’adapté selon les besoins, car je pense qu’il serait d’une grande utilité pour le CFPT-I[[12]](#footnote-12).

Améliorations possibles :

* Ajouter un sel dans le hachage du mot de passe.
* Les évaluations et les notes étant des données « sensibles », les encrypter avec un algorithme de chiffrement symétrique (comme AES[[13]](#footnote-13))
* Pouvoir ajouter une image au projet
* Pouvoir rajouter des fichiers dans les projets pour sauvegarder le travail fourni
* Ajouter une gestion centralisée des notes

## Comparaison planning effectif

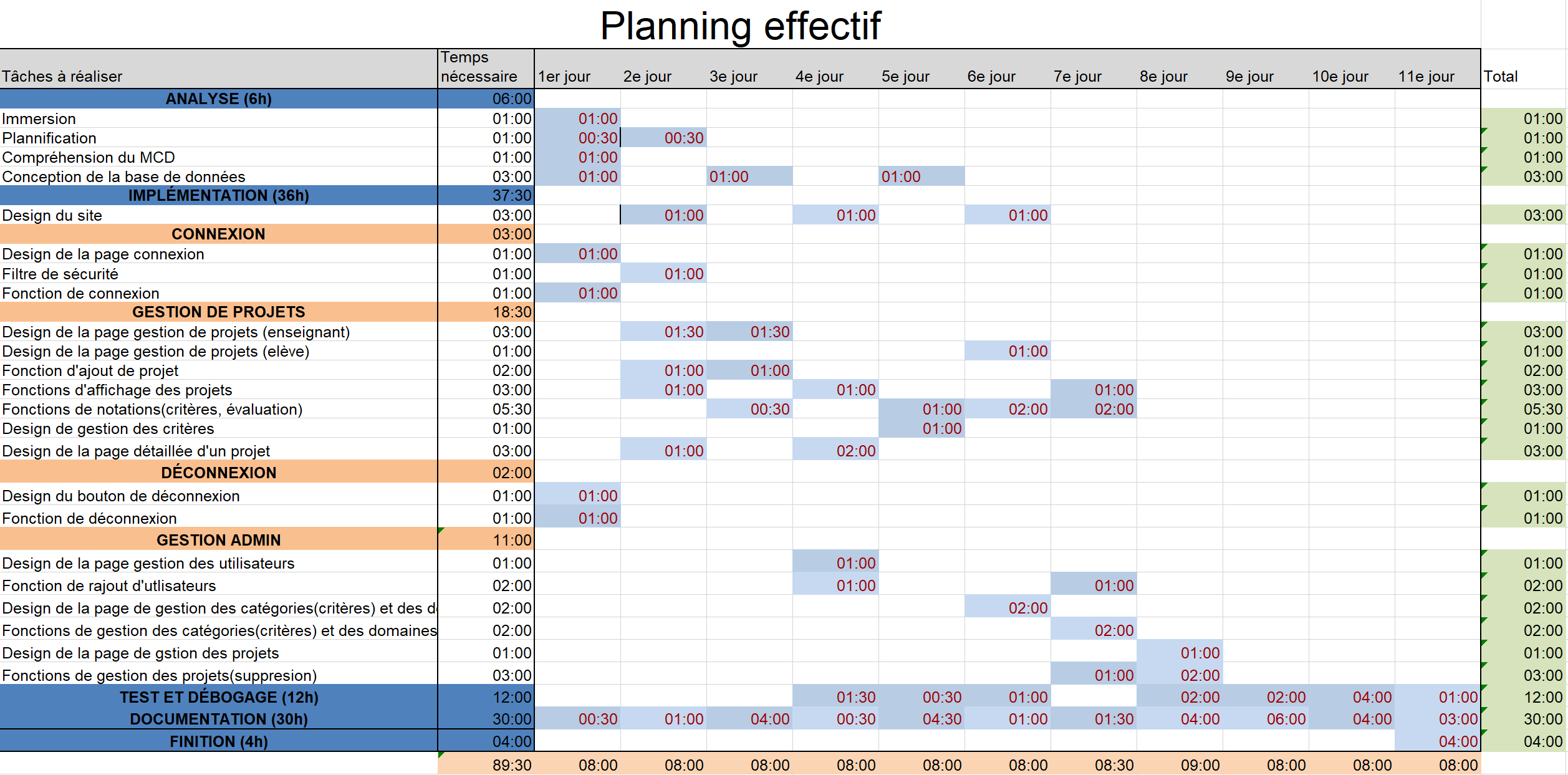


Figure 36 Planning effectif

Le planning effectif est assez différent du planning prévisionnel car je ne me rendais pas compte au début du TPI, mais je dois souvent revenir sur les fonctionnalités précédentes pour en apporter des modifications. Au niveau de la documentation et des tests, on remarque que la programmation en est séparée (au niveau des jours). Ceci est dû au fait que je pense que ma concentration est plus élevée si je travaille sur le même domaine (programmation, écriture, planification).

## Documents à rendre

* Documentation technique
* Journal de bord
* Manuel utilisateur
* Résumé du rapport du TPI
* Code source (PDF)
* Fichiers de l’application (code source)

# Bibliographie

## Sites d’aide à la programmation

<https://stackoverflow.com>

<https://getbootstrap.com>

<https://www.w3schools.com>

<https://php.net>

<https://paletton.com>

https://wikipedia.com

## Hébergement sauvegarde

<https://github.com>

<https://drive.google.com>

# Annexes

## Manuel utilisateur

Voir fichier AdrienWitzig\_ManuelUtilisateur.pdf

## Résumé du TPI

Voir fichier Resume\_Witzig\_EvaluationEE.pdf

## Planification

Voir fichier AdrienWitzig\_Planification.xlsx

## Journal de bord

Voir fichier journal-de-bord\_AdrienWitzig.pdf

## Code source de l’application

Voir fichier AdrienWitzig\_codeSource.pdf

# Table des illustrations

[Figure 1 Modèle conceptuel de données fourni pour l'énoncé 5](#_Toc42422752)

[Figure 2 Lors d’une ouverture d’une fenêtre modal en PHP 7](#_Toc42422753)

[Figure 3 Lors d'ajout d'utilisateur, si l'administrateur ajoute un élève, les champs "classe" et "année" apparaissent 7](#_Toc42422754)

[Figure 4 Fenêtre de connexion 9](#_Toc42422755)

[Figure 5 Page "Mes projets" (pour élèves) 10](#_Toc42422756)

[Figure 6 Page gestion de projets (pour enseignants) 10](#_Toc42422757)

[Figure 7 Modal d’ajout projets (pour enseignants) 11](#_Toc42422758)

[Figure 8 Page de détail d'un projet (pour élèves) partie 1 12](#_Toc42422759)

[Figure 9 Page de détail d'un projet (pour élèves) partie 2 12](#_Toc42422760)

[Figure 10 Page "gestion de projets" (pour enseignants) partie 1 avec élève évalué 13](#_Toc42422761)

[Figure 11 Page "gestion de projets" (pour enseignants) partie 2 avec élève évalué 13](#_Toc42422762)

[Figure 12 Page "gestion de projets" (pour enseignants) sans aucun élèves évalués 14](#_Toc42422763)

[Figure 13 Modal de modification d'un projet (pour enseignants) 15](#_Toc42422764)

[Figure 14 Modal pour la suppression de projet (pour enseignants) 16](#_Toc42422765)

[Figure 15 Modal d'ajout d'élèves dans un projet (pour enseignants) 16](#_Toc42422766)

[Figure 16 Modal d'ajout de critères dans un projet (pour enseignants) 17](#_Toc42422767)

[Figure 17 Page "voir évaluation" (pour enseignants) partie 1 18](#_Toc42422768)

[Figure 18 Page "voir évaluation" (pour enseignants) partie 2 18](#_Toc42422769)

[Figure 19 Page d'évaluation d'un élève (pour enseignants) 19](#_Toc42422770)

[Figure 20 Page de gestion des critères (pour enseignants et administrateurs) 19](#_Toc42422771)

[Figure 21 Modal d'ajout d'un nouveau critère 20](#_Toc42422772)

[Figure 22 Page profil 20](#_Toc42422773)

[Figure 23 Modal de changement de mot de passe 21](#_Toc42422774)

[Figure 24 Gestion administrateur 21](#_Toc42422775)

[Figure 25 Gestion des projets (administrateur) 22](#_Toc42422776)

[Figure 26 Architecture du code du site 23](#_Toc42422777)

[Figure 27 Requête d'affichage des élèves disponibles pour un projet 25](#_Toc42422778)

[Figure 28 Script de déconnexion 25](#_Toc42422779)

[Figure 29 planning effectif 36](#_Toc42422780)

# Tableaux

[Tableau 1 Tests 34](#_Toc42422692)

[Tableau 2 Tests 35](#_Toc42422693)

1. En format .PDF [↑](#footnote-ref-1)
2. Variable de session PHP : variables PHP multipages non-cryptées [↑](#footnote-ref-2)
3. modal : Elément de la libraire Bootstrap sous forme de fenêtre pop-up (voir 3.3) [↑](#footnote-ref-3)
4. CDN : Réseau de diffusion de contenu [↑](#footnote-ref-4)
5. responsive : Qui s’adapte selon l’appareil (Ordinateurs, Tablettes, Smartphones, etc…) [↑](#footnote-ref-5)
6. <button> : Balise HTML du bouton [↑](#footnote-ref-6)
7. modal : Elément de la libraire Bootstrap sous forme de fenêtre pop-up (voir 3.3) [↑](#footnote-ref-7)
8. Sha1 : algorithme de hachage permettant d’obtenir une « empreinte » de données, en principe, unidirectionnel. [↑](#footnote-ref-8)
9. Variable de session PHP : variables PHP multipages non-cryptées [↑](#footnote-ref-9)
10. Robustesse dans une application : Si l’application réagit de manière adaptée aux actions utilisateurs. [↑](#footnote-ref-10)
11. <https://www.php.net/manual/fr/function.utf8-decode.php> [↑](#footnote-ref-11)
12. CFPT-I : Département informatique du CFPT [↑](#footnote-ref-12)
13. AES : *Advanced Encryption Standard :* algorithme de chiffrement symétrique [↑](#footnote-ref-13)