

# Compte rendu d'activité

---

Projet : gestion des vacataires

## **Table des matières :**

<b>1. Introduction</b>	<b>3</b>
<b>2. Mes tâches</b>	<b>3</b>
2.1. Page d'authentification	3
2.2. Page de profil	3
<b>3. Conclusion</b>	<b>3</b>

# 1. Introduction

Le nombre important de vacataires présents à l'IUT pose un problème d'organisation pour les enseignants responsables. À l'heure actuelle, aucune solution spécifique ne permet de résoudre ce problème. Dans ce contexte, une équipe d'étudiants de troisième année en formations initiale a été chargée de développer une application web de gestion des vacataires.

L'application était fonctionnelle lors de l'arrivée de l'équipe de renforcement, mais elle manquait de certaines fonctionnalités essentielles, telles qu'une page d'authentification, des filtres pour les modules et les vacataires, et une protection de l'API. Les tâches nécessaires à la mise en place de ces fonctionnalités n'avaient pas été formalisées sur GitHub, et la répartition des tâches s'est faite à l'oral. La documentation était également insuffisante, ce qui a rendu difficile la compréhension du projet. De plus, le développement se déroulait dans un environnement de production, ce qui le rendait compliqué.

## 2. Mes tâches

### 2.1. Page d'authentification

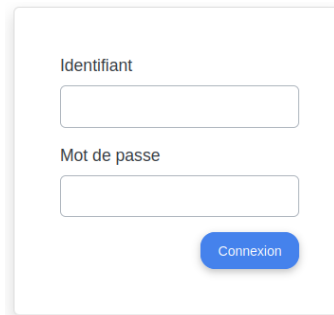
Avec Bryce Fuertes, nous nous sommes chargés de mettre en place un système d'authentification pour l'application. L'objectif était de restreindre l'accès au site uniquement aux utilisateurs connectés. Pour ce faire, il fallait créer une page de connexion sur lequel les utilisateurs non connectés seraient redirigés ainsi qu'enregistrer un mot de passe en base de données de manière sécurisée afin de pouvoir assurer la connexion.

Nous avons tout d'abord créé une branche "dev-auth" sur GitHub à partir de la branche "dev" pour pouvoir travailler sans créer de conflits avec les autres développements en cours. Nous avons ensuite créé une page de connexion. À partir de ce moment, deux solutions se sont présentées à nous : utiliser le portail de connexion de l'université (CAS) ou créer notre propre système de connexion. N'ayant pas trouvé de documentation et ayant peur d'une intégration complexe, nous avons abandonné la solution du CAS. Nous avons donc fait le choix de créer notre propre système de connexion.

Nous avons réalisé un simple formulaire de connexion permettant de rentrer un identifiant et un mot de passe. Une fois ces identifiants récupérés, nous avons créé une requête HTTP dans l'API permettant de traiter cette connexion.

Ainsi, dans l'API, le mot de passe en clair était comparé à un mot de passe préalablement salé et hashé enregistré en base de données. Pour effectuer cette comparaison, nous avons utilisé la librairie bcrypt.

Une fois l'ensemble de ces éléments mis en place, il était possible de se connecter. Bryce Fuertes et Michele Florio se sont ensuite chargé de la redirection dans le cas d'un utilisateur non connecté et de la sécurisation des requêtes à l'API à l'aide d'un JSON Web Token

A Bootstrap-style login form. It features two text input fields: the first is labeled 'Identifiant' and the second is labeled 'Mot de passe'. Below the second field is a blue button with the text 'Connexion' in white.

*Formulaire de connexion réalisé en bootstrap*



*Issues github lié à la tâche réalisée*

## 2.2. Page de profil

Je me suis chargé de réaliser une page de profil en complément de la page de connexion. Cette page devait permettre à l'utilisateur, une fois connecté, de pouvoir changer de mot de passe et de se déconnecter.

Travaillant seul sur cette tâche, je n'ai pas créé de nouvelles branches. J'ai travaillé en local à partir d'un clone de la branche "dev".

J'ai commencé par créer une nouvelle page "profil". Sur cette page, j'ai réalisé un formulaire de changement de mot de passe comprenant deux champs : l'ancien mot de passe et le nouveau mot de passe voulu. J'ai aussi ajouté un bouton de déconnexion.

J'ai commencé par développer le bouton de déconnexion dont le fonctionnement est assez simple. L'enregistrement local de l'utilisateur et vider, supprimant ainsi le jeton d'accès permettant d'accéder aux requêtes et aux différentes pages du site. Dans un second temps, l'utilisateur est automatiquement redirigé vers la page de connexion.

Par la suite, je me suis occupé du changement de mot de passe. Une fois les données du formulaire récupéré, une requête est effectuée à l'API. Celle-ci réalise une fonction testant si le mot de passe actuel est correct et, s'il l'est, enregistre une version salée et hashé du nouveau mot de passe en base à la place de l'ancien.



*Issues github correspondants à mes réalisations*

### 3. Conclusion

Lors de ces deux semaines de soutiens sur le projet des élèves en formation initiale, je suis parvenu à réaliser l'ensemble des tâches qui m'étaient attribuées et plus encore avec notamment la fonctionnalité de changement de mot de passe. Malgré une certaine désorganisation de l'équipe, la répartition des tâches efficaces dès notre arrivée sur le projet a permis à chaque membre de travailler efficacement sans ralentir le reste des coéquipiers. Cette période est pour moi une réussite ayant atteint l'ensemble des objectifs que je m'étais fixé et qui m'avait été donné par mon équipe de travail.

Pour des développements futurs, je pense qu'une meilleure gestion des erreurs pourraient être imaginés. Une gestion de compte plus avancé et la création de compte multiple pourrait aussi être envisagé, un seul compte étant présent sur la version actuelle du site.

Je tiens à remercier tous les membres de mon équipe pour cette période à l'IUT. Je remercie Bryce Fuertes pour sa collaboration, Michele Florio pour son aide précieuse et Marco Valle pour avoir mis en place un docker permettant de déployer un environnement de développement en local facilitant le développement.