

PLATEFORME WEB POUR LE TRAITEMENT DES DOSSIERS DES CANDIDATS POUR LA MAÎTRISE EN INFORMATIQUE

Mariane Pouliot (étudiante) et Yacine Yaddaden

Département de mathématiques, informatique et génie, Université du Québec à Rimouski

Problématique

Afin d'étudier à la maîtrise en informatique, les candidats doivent solliciter un professeur afin que ce dernier accepte de superviser leur travail de recherche. Toutefois, n'ayant pas toujours connaissance des documents à fournir et les intérêts de recherche de chacun des professeurs, ils envoient généralement des courriels à l'ensemble des professeurs. Cela engendre certains problèmes :

- ✗ Le dossier est incomplet,
- ✗ Plusieurs courriels sont échangés,
- ✗ Plusieurs professeurs donnent leurs accords pour la direction du même étudiant,
- ✗ Des professeurs sont contactés alors que l'intérêt de recherche et profil de l'étudiant ne correspondent pas à leurs domaines d'expertise.

Description de la plateforme

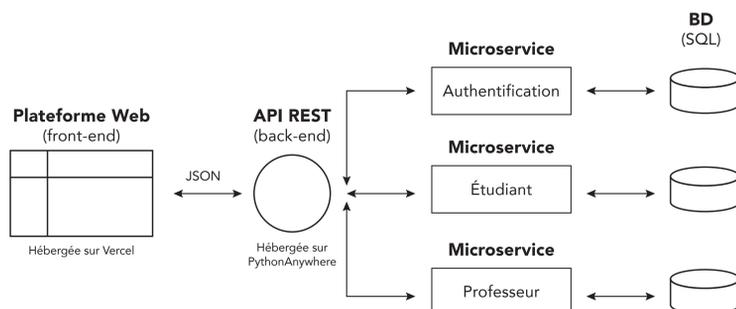


FIGURE 1 – Fonctionnement de la plateforme

Solution

Le développement d'une plateforme Web en trois parties permettra de faciliter et d'automatiser le traitement des dossiers des candidats.

- Une première partie pour les étudiants, afin qu'ils soumettent leur candidature.
- Une seconde partie pour les professeurs afin qu'ils puissent voir les candidatures déposées et signifier leurs intérêts.
- Une troisième partie pour l'administrateur afin de gérer la plateforme (*jumelage de professeurs et d'élèves, ajouter un professeur, supprimer des candidatures, etc.*).

Back-end – côté serveur

L'utilisation d'une architecture sous forme de **microservices** [1] dans le **back-end** comporte plusieurs avantages. Voici un tableau les récapitulant de manière non exhaustive :

Avantage	Explication
PERFORMANCE	Les microservices divisent les applications volumineuses en partie indépendantes, offrant ainsi aux développeurs plus de flexibilité pour créer des applications qui sont légères et performantes.
FACILITÉ	La création d'une application avec des microservices réduit la complexité et facilite grandement la maintenance, car la modification de petits éléments peut se faire individuellement, et ce sans affecter le reste de l'application.
RÉSILIENCE	Comme chaque service est indépendant, aucun n'a d'impact sur un autre, ce qui signifie que si l'un tombe en panne, les autres resteront en place.
SÉCURITÉ ACCRUE	Les microservices permettent la séparation des données à l'aide de la mise en place de plusieurs BD. Possédant ainsi plusieurs bases de données, il est plus difficile pour les pirates de compromettre l'application.

TABLE 1 – Avantage de l'utilisation des microservices

Voici la liste des technologies utilisées afin de développer le **back-end** :

Langages de programmation → Python

Frameworks et bibliothèques → Django (Django REST Framework + SWAGGER)

Environnements → Python (*environnement virtuel*)

Éditeur de code → Microsoft Visual Studio Code

Front-end – côté client

Voici la liste des technologies utilisées afin de développer le **front-end** :

Langages de programmation → JavaScript (ES6) avec HTML5/CSS3.

Frameworks et bibliothèques → Twitter Bootstrap 5, ReactJS (React-Router + React-Query), Axios

Environnements → Node.js (LTS)

Éditeur de code → Microsoft Visual Studio Code

React-Query [2] est une bibliothèque de gestion des données qui permet de récupérer, de mettre en cache, de synchroniser et de mettre à jour l'état du serveur d'une application développée sous React.

Il tire son principal avantage de sa **simplicité à résoudre les problèmes liés à la gestion asynchrone des données**. En effet, grâce à son utilisation :

- Il n'y a pas d'état global à gérer, de réducteurs, de systèmes de normalisation ou de configurations lourdes à comprendre.
- Il s'agit simplement de passer une fonction (*mentionnant l'endroit de la récupération des informations et à quel point l'utilisation sera requise*) qui résout les données ou génère une erreur.

Aperçu du visuel

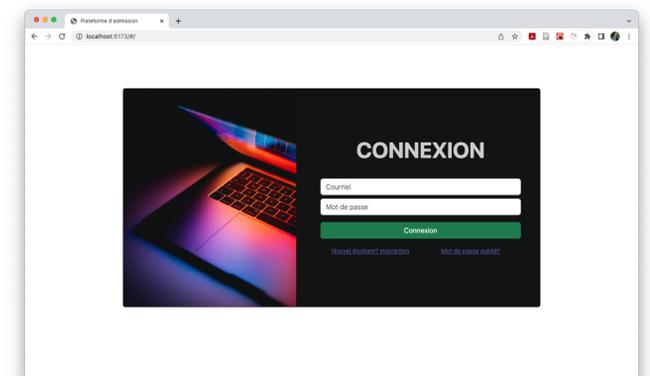


FIGURE 2 – Page de connexion

Bilan

Le développement de cette plateforme permet :

- ✓ L'amélioration du processus de demande et d'admission à la maîtrise en informatique (*gestion des dossiers*).
- ✓ Un généralisation de ce processus dans plusieurs départements de l'Université.

Références

[1] Akos Hochrein. *Designing Microservices with Django : An Overview of Tools and Practices*. Apress, 2019.

[2] React-Query. Tanstack query | react query, solid query, svelte query, vue query. <https://tanstack.com/query/v3/>, 2023.