



FreeJobs

# RAPPORT DE STAGE


| MILED Yassine  
| BTS SIO SLAM

du 2 décembre au 26 janvier





# Sommaire

- **PAGE 3 : Présentation de l'entreprise**
  - **PAGE 4 : Organisation au sein de l'entreprise**
  - **PAGE 6 : Mes missions**
  - **PAGE 10 : ANNEXES**
  - **PAGE 12 : Conclusion**
- 

# Présentation de l'entreprise FreeJobs

FreeJobs est une plateforme qui facilite la recherche d'emploi et le recrutement. Elle permet aux candidats de postuler directement aux offres grâce à des formulaires simples, sans avoir besoin de créer un compte.

Les recruteurs peuvent aussi publier facilement des offres d'emploi pour toucher un large public.

L'entreprise est dirigée par Amine et Clément, qui ont pour objectif de rendre le recrutement simple et accessible à tous.



# Organisation au sein de l'entreprise

- Encadrement : Mon stage a été supervisé par Clément en tant que tuteurs, sous la direction de Lotfi, chef de projet.
- Méthodologie : L'organisation du projet s'est effectuée selon une approche Agile, structurée en sprints dont la durée était définie au début de chaque sprints.
- Gestion des tâches : Les différentes tâches étaient gérées via l'outil Jira, chaque ticket (tâche) étant attribué à un participant du projet.
- Suivi du projet : Des réunions quotidiennes (daily meetings) étaient organisées pour évaluer l'avancement des tâches et résoudre d'éventuels blocages.
- Livrables : Une démonstration des résultats obtenus était présentée à la fin de chaque sprint.
- Utilisation de technologies : La base de données NoSQL MongoDB a été utilisée pour stocker et gérer les données, offrant flexibilité et performances adaptées aux besoins du projet.

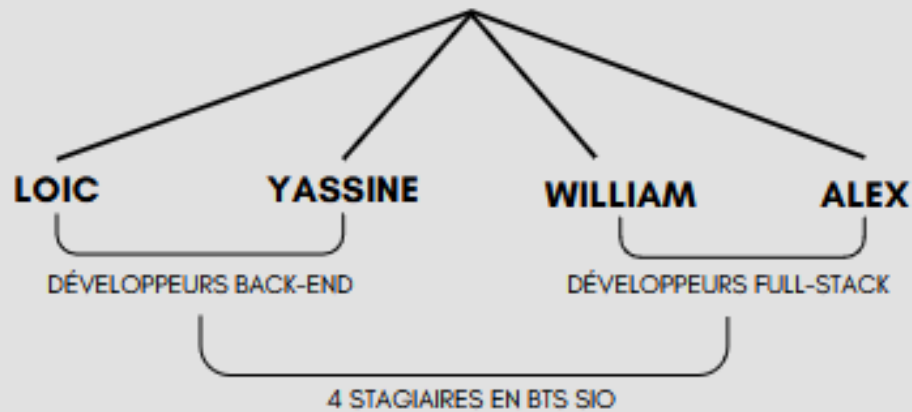


# ORGANIGRAMME

## LOTFI

CHEF DE PROJET

*DEVELOPPEUR BAC+5*



# Mes missions

## Création d'un modèle Application

Lors de mon premier sprint d'une durée d'une semaine, j'ai travaillé sur la création d'un modèle Application (candidature). Ce modèle est lié aux modèles Job et Candidat, qui ont été créés par deux autres stagiaires. Le modèle Application récupère les identifiants CandidatID et JobID afin de permettre la création d'une nouvelle candidature.

Cette tâche permet de structurer les données et faciliter la gestion des candidatures au sein de l'application.

```
TS application.ts  
TS blocklistModel.ts  
TS candidate.ts  
TS job.ts
```

## Connexion avec l'API CRM

Lors de mon deuxième sprint d'une durée de deux semaines, j'ai travaillé sur l'intégration d'une connexion avec l'API CRM. Cette tâche comprend deux objectifs principaux :

- Récupérer la liste des communautés : Tirer automatiquement la liste des communautés disponibles depuis l'API CRM afin de les synchroniser avec l'application.
- Ajouter un nouveau candidat dans le CRM : Lorsqu'une nouvelle candidature est créée dans l'application, le système doit ajouter automatiquement le candidat correspondant dans le CRM.

Cette mission, vise à assurer une meilleure intégration et synchronisation des données entre l'application et le CRM.

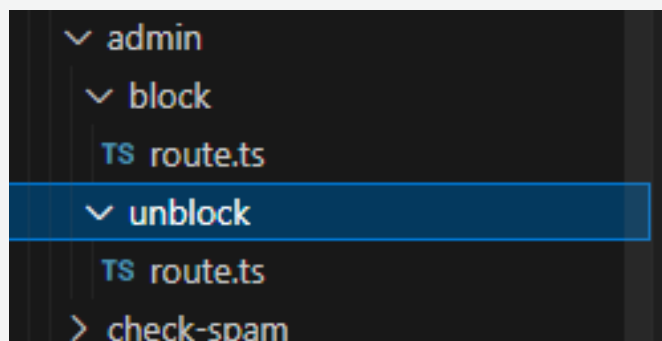
```
export const addCandidateToCRM = async (candidateData: {  
  firstName: string;  
  lastName: string;  
  community: string[];  
  email: string;  
  linkedin: string;  
}) => {  
  const response = await fetch(`${process.env.NEXT_PUBLIC_CREATE_CONTACT}/api/contacts/create`  
    , {  
      method: 'POST',  
      headers: {  
        'Content-Type': 'application/json',  
        'external-muko': 'crm'  
      }  
    })
```

## Mise en place d'un système de blocage des IPs/machines identifiées comme indésirables

Dans le cadre de cette tâche, j'ai implémenté un système pour bloquer les adresses IP ou les machines identifiées comme sources de spam. Cette tâche comprend les éléments suivants :

- Création d'un modèle BlockedList : Stocker les adresses IP ou identifiants de machines indésirables.
- Ajout d'un middleware : Vérifier chaque requête entrante en la comparant à la liste des IPs bloquées.
- Mécanisme de déblocage manuel : Permettre la gestion manuelle des IPs/machines bloquées.
- Les IPs/machines bloquées ne peuvent pas accéder aux endpoints de l'API.

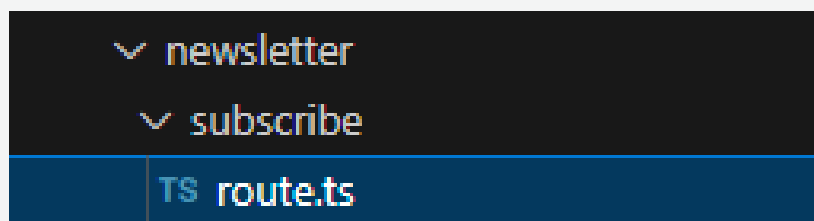
Cette fonctionnalité a renforcé la sécurité de l'application en limitant les risques liés au spam ou à d'autres comportements malveillants.





## **Notifier les utilisateurs des nouvelles offres d'emploi dans leurs communautés par email. (Newsletter)**

- Implémenter une fonctionnalité permettant aux utilisateurs de s'abonner à des newsletters.
- Créer une tâche planifiée pour envoyer des notifications hebdomadaires sur les nouvelles offres d'emploi aux abonnés.
- Ajouter une option permettant aux utilisateurs de se désabonner.
- Les utilisateurs reçoivent des emails contenant des offres d'emploi pertinentes pour leur communauté.
- Les informations de l'utilisateur doivent être enregistrées dans la base de données.



# ANNEXES

Projets / FreeJobs\_Platform / Ajouter epic / JOBS-350

Block identified spam IPs or machines.

+ Ajouter

Description

Description:

Maintain a BlockedList model storing IPs or machine identifiers.

Add middleware to check each request against the blocked list.

Provide a mechanism to unblock IPs manually.

Acceptance Criteria:

Blocked IPs/machines cannot access any API endpoints.

Activité

Tout

Commentaires

Historique

YM

Ajouter un commentaire...

Joli !

Besoin d'aide ?

C'est bloqué...

Conseil de pro : appuyez sur M pour commenter

Terminer

Terminé

Détails

Personne assignée

YM Yassine Miled

Étiquettes

Aucun

Parent

Aucune

Sprint

Aucun +1

Story point estimate

Aucun

Développement

Créer une branche

Créer un commit

Rapporteur

LOTFI JBELI

Automatisation

Exécutions de règles

Création 23 décembre 2024 à 08:19

Mis à jour 16 janvier 2025 à 08:13

Résolu 16 janvier 2025 à 08:13

Branch	Updated	Check status	Behind	Ahead	Pull request
dev	5 days ago		0	258	
357-Yassine	2 weeks ago		0	233	#31
361-William	2 weeks ago		0	238	#35
revert-32-358-Alex	2 weeks ago		0	232	
358-Alex	2 weeks ago		0	229	#34

```
lib > TS mongodb.ts > ...
1  const mongoose = require("mongoose");
2
3  const { MONGODB_URI } = process.env;
4  export const connectDB = async () => {
5    try {
6      const { connection } = await mongoose.connect(MONGODB_URI as string);
7      if (connection.readyState === 1) {
8        return Promise.resolve(true);
9      }
10   } catch (error) {
11     console.error(error);
12     return Promise.reject(error);
13   }
14 }
```

```

_models > TS application.ts > ...
1  import mongoose, { Schema, model, Model } from "mongoose";
2
3  interface Application {
4    _id: mongoose.Types.ObjectId;
5    candidateID: mongoose.Types.ObjectId;
6    jobID: mongoose.Types.ObjectId;
7  }
8
9  const ApplicationSchema = new Schema<Application>({
10   {
11     candidateID: {
12       type: Schema.Types.ObjectId,
13       ref: "Candidate",
14       required: [true, "L'identifiant du candidat est obligatoire."],
15     },
16     jobID: {
17       type: Schema.Types.ObjectId,
18       ref: "Job",
19       required: [true, "L'identifiant du poste est obligatoire."],
20     },
21   },
22   { timestamps: true }
23 });

```

MongoDB Compass - FreeJobs/FreeJobs.blockedlists

Connections Edit View Collection Help

Compass

() My Queries

CONNECTIONS (1)

Search connections

FreeJobs

- FreeJobs
  - applications
  - blockedlists
  - candidates
  - jobs
  - newsletters
  - spamlogs
  - subscribers
  - unverifiedjobs
- admin
- config
- local
- perff

FreeJobs > FreeJobs > blockedlists

Documents 1 Aggregations Schema Indexes 3 Validation

Type a query: { field: 'value' } or [Generate query](#)

Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE

25 1-1 of 1

```

_id: ObjectId('67921b9ce70d36caa57aac15')
ipAddress: "::1"
_v: 0
blockedAt: 2025-01-23T10:36:12.708+00:00
blockedUntil: 2025-01-23T11:36:12.425+00:00
notes: "Blocage automatique suite à la détection de spam"
reason: 1

```

CRM

perff

- businesshistories
- communities
- companies
- contacthistories
- contacts
- contactsbusinesses
- lists
- users

Compass 1.46.1 installed successfully

Release Notes

## Conclusion

L'apprentissage de Next.js a été une expérience enrichissante qui m'a permis de renforcer mes compétences en développement web moderne, notamment grâce à la création de nouvelles fonctionnalités. Ce projet a également mis en lumière l'importance du travail d'équipe, où la collaboration et la coordination ont joué un rôle clé dans le succès des différentes étapes.

De plus, l'utilisation d'outils de gestion de projet comme Jira m'a permis d'organiser et de suivre efficacement les tâches, tandis que l'usage de GitHub (pull requests, commits, branches) a renforcé mes bonnes pratiques en gestion de version et en travail collaboratif.

Ces expériences m'ont permis de développer des compétences techniques et organisationnelles précieuses, applicables à des projets futurs.