

# Documentation complète du projet MIRAI

---

## Présentation du projet

### Membres de l'équipe

- **Sabrina Eloundou**
- **Johan Tichit**

### Titre du projet

#### **MIRAI — La plateforme RH intelligente pour freelances et entreprises**

### Contexte

Ce projet a été réalisé dans le cadre de l'UF "Développement Logiciel et Base de Données" du Bachelor 2 à Ynov Campus. L'objectif pédagogique est de mettre en œuvre toutes les compétences acquises dans les domaines de la programmation orientée objet, des bases de données relationnelles, de la conception logicielle, et de l'architecture logicielle moderne. Le thème du projet est libre, à condition de répondre à des critères fonctionnels et techniques précis.

### Objectifs

- Créer une marketplace intuitive et sécurisée permettant la mise en relation entre entreprises et freelances.
- Offrir un système de matching intelligent basé sur les compétences, expériences et critères spécifiques des missions.
- Mettre en place un système de paiement sécurisé avec Stripe Connect.
- Simplifier la gestion des missions, candidatures, messagerie et notations.
- Proposer une UX fluide sans complexité technique pour les utilisateurs.

### Résumé global

MIRAI est une plateforme web moderne où les freelances peuvent trouver des missions adaptées à leur profil, et où les entreprises peuvent facilement publier des offres et entrer en contact avec les meilleurs talents. Le système intègre un score de matching intelligent, une messagerie intégrée, un système d'évaluation, ainsi qu'un module de paiement automatisé et sécurisé via Stripe Connect. Le tout dans une interface sobre, efficace et responsive.

## Conception du projet

### Analyse des besoins

- Simplifier la recherche de missions pour les freelances.
- Automatiser la gestion des candidatures pour les entreprises.
- Éviter les allers-retours externes pour les paiements.
- Offrir une interface claire et des interactions rapides.

## 🔧 Choix techniques et outils utilisés

Les technologies ont été sélectionnées selon plusieurs critères : compatibilité avec les compétences de l'équipe, performance, sécurité, facilité de développement, documentation disponible et bonne intégration entre elles.

- **Frontend** : Next.js (React Framework) pour une structure robuste côté client, CSS Classique & Tailwind CSS une gestion rapide et personnalisée des styles.
- **Backend** : Node.js et Express.js pour un serveur léger et rapide, Prisma comme ORM moderne pour interagir facilement avec la base PostgreSQL.
- **Base de données** : PostgreSQL, base robuste et relationnelle, adaptée aux besoins du projet (relations multiples, intégrité des données).
- **Paiement** : Stripe Connect (mode Custom) pour permettre aux freelances de recevoir directement les paiements sur leur compte bancaire sans créer de compte Stripe visible.
- **Authentification** : JSON Web Tokens (JWT) pour sécuriser les sessions utilisateurs et restreindre l'accès selon les rôles (freelance / entreprise).
- **Déploiement** : Serveur en machine virtuelle (VM) personnelle, permettant le contrôle complet de l'environnement et la maîtrise des déploiements.
- **Versioning** : Git pour la gestion des versions du code source, et GitHub pour le travail collaboratif, les pull requests et l'historique des commits.

## 🏗 Architecture / structure du projet

- Base de données relationnelle modélisée autour des entités : **users**, **profiles**, **missions**, **applications**, **messages**, **transactions**, **reviews**, **matching**.
- Séparation claire des rôles : entreprise vs freelance.
- Architecture REST côté backend.
- Architecture modulaire côté frontend avec des layouts dédiés (vitrine/extranet).

## 🔗 Fonctionnalités principales

- Authentification sécurisée avec JWT.
- Création, édition et suppression de missions.
- Postuler à une mission et suivre l'état des candidatures.
- Matching intelligent basé sur des critères pondérés (compétences, budget, localisation).
- Système de messagerie intégré.
- Paiement sécurisé via Stripe vers le compte du freelance.
- Notation et avis sur les utilisateurs après mission.
- Tableau de bord personnalisé pour freelance et entreprise.

## 📌 Gestion du projet

### 👥 Organisation de l'équipe

- **Sabrina Eloundou** : Développement du backend, base de données, architecture Stripe, sécurité.
- **Johan Tichit** : Développement du frontend, interface utilisateur, intégration API, composants React.

### 📅 Planification

- **12 mars** : Formation des binômes et dépôt du sujet.
- **15–24 mars** : Mise en place de la BDD, installation des outils, architecture initiale.
- **25 mars – 2 avril** : Authentification, gestion utilisateur, CRUD missions.
- **3 avril – 10 avril** : Matching, messagerie, système d'avis.
- **10 – 20 avril** : Paiement Stripe, tests, documentation.
- **20 – 30 avril** : Ajustements finaux, responsive, accessibilité.
- **1 – 10 mai** : Finalisation de la documentation, VM, README, livrables.

## 🌀 Méthodologie de travail

- Approche itérative inspirée de l'agilité (sprints hebdo).
- Synchronisations régulières entre membres (Discord).
- Partage de l'avancement via Git et branches fonctionnelles.

## 🗒 Répartition des tâches et difficultés rencontrées

- Le plus gros challenge a été l'intégration de Stripe Connect sans complexifier l'expérience freelance (objectif : éviter les inscriptions externes).
- Mise en place du système de matching avec pondération dynamique.
- Organisation claire du code pour faciliter la contribution du binôme (notamment côté frontend).
- Synchronisation entre frontend/backend au fur et à mesure des avancées.
- L'intégration des formulaires Stripe et leur sécurisation a demandé un vrai effort de test.

---

📌 **Ce document servira de guide pour organiser le développement de MIRAI.**

Fait par : @MIRAI\_DEV