



Licence MIAGE — Troisième année

InfluMatch

Plateforme de mise en relation entre influenceurs et entreprises

Développement Web

Membres du groupe

HAMMOUDA Nourane

BENMAHMOUD Zaineb

MABROUKI Chaïma

MOTHU Ornela

18 décembre 2025

Dépôt GitHub :

github.com/nourane-hammouda/InfluMatch

Table des matières

1	Introduction	4
1.1	Présentation du Projet	4
1.2	Objectifs du Projet	4
1.3	Technologies Utilisées	4
1.3.1	Backend	4
1.3.2	Frontend	5
1.3.3	Outils et Infrastructure	5
2	Architecture du Projet	6
2.1	Architecture Générale	6
2.1.1	Architecture MVC Adaptée	6
2.2	Structure des Répertoires	6
2.2.1	Backend (Django)	6
2.2.2	Frontend (React)	7
2.3	Communication Backend-Frontend	7
3	Modèles de Données	8
3.1	Architecture de la Base de Données	8
3.2	Modèle User	8
3.2.1	Relations	8
3.3	Modèle Influenceur	8
3.3.1	Informations de Base	8
3.3.2	Métriques et Statistiques	8
3.4	Modèle Entreprise et Campagne	9
3.4.1	Modèle Entreprise	9
3.4.2	Modèle Campagne	9
3.5	Modèle Candidature	9
3.5.1	Caractéristiques	9
3.6	Modèle Notification	10
4	Système d'Authentification	10
4.1	Architecture JWT	10
4.2	Processus d'Inscription et de Connexion	10
4.3	Gestion et Sécurité des Tokens	10
4.4	Protection des Routes	11

5	Gestion des Profils	11
5.1	Architecture de Gestion des Profils	11
5.2	Profil Influenceur	11
5.2.1	Composants du Profil	11
5.2.2	Complétion du Profil	11
5.3	Mise à Jour du Profil	11
5.4	Récupération des Informations Utilisateur	12
6	Interface Utilisateur Frontend	12
6.1	Architecture Frontend	12
6.2	Pages Principales	12
6.2.1	Page d'Accueil	12
6.2.2	Pages d'Authentification	12
6.2.3	Dashboard	12
6.2.4	Marketplace	13
6.2.5	Page de Détails d'Offre	13
6.2.6	Gestion des Candidatures	13
6.2.7	Profil Utilisateur	13
6.2.8	Complétion du Profil	14
6.3	Composants Réutilisables	14
6.4	Gestion d'État	14
6.5	Routing et Navigation	14
6.6	Service API	14
6.6.1	Architecture du Service	14
6.6.2	Endpoints API	15
7	Endpoints API	15
7.1	Architecture REST	15
7.2	Endpoints d'Authentification	15
7.3	Endpoints de Profil	15
7.4	Gestion des Erreurs	15
8	Fonctionnalités Techniques	16
8.1	Système de Signaux Django	16
8.2	Interface d'Administration Django	16
8.3	Configuration CORS	16
8.4	Configuration de la Base de Données	16
8.4.1	MySQL Connector	16
8.4.2	Migrations Django	17
8.5	Sécurité	17
8.5.1	Protection CSRF	17

8.5.2	Validation des Données	17
8.5.3	Hashage des Mots de Passe	17
9	Base de Données	17
9.1	Schéma Relationnel	17
9.2	Tables Principales	17
9.2.1	Utilisateurs	17
9.2.2	Influenceurs	18
9.2.3	Entreprises	18
9.2.4	Campagnes	18
9.2.5	Candidatures	18
9.3	Tables de Liaison	18
9.3.1	Domaines d'Expertise	18
9.3.2	Plateformes Sociales	18
9.4	Relations et Intégrité	19
9.4.1	Types de Relations	19
9.4.2	Contraintes d'Intégrité	19
9.5	Optimisations	19
9.5.1	Index	19
9.5.2	Requêtes Optimisées	19
10	Configuration et Déploiement	19
11	Conclusion	19
11.1	Améliorations Futures	20
11.2	Apprentissages et Défis	20
12	Webographie	21
13	Annexes	21

1 Introduction

1.1 Présentation du Projet

InfluMatch est une plateforme web complète permettant la mise en relation entre influenceurs et entreprises pour des collaborations publicitaires. Le projet a été développé avec une architecture full-stack moderne, combinant Django REST Framework pour le backend et React avec TypeScript pour le frontend.

Le code source du projet est disponible sur GitHub : <https://github.com/nourane-hammouda/InfluMatch>

La plateforme offre un écosystème complet où les influenceurs peuvent créer et gérer leur profil professionnel, consulter des offres de campagnes publicitaires, et postuler aux opportunités qui correspondent à leur domaine d'expertise. Pour les entreprises, la plateforme permet de publier des campagnes et de découvrir des influenceurs adaptés à leurs besoins marketing.

1.2 Objectifs du Projet

Les objectifs principaux du projet sont :

- **Authentification sécurisée** : Mettre en place un système d'authentification robuste avec JWT (JSON Web Tokens) permettant la gestion des sessions utilisateurs de manière sécurisée
- **Gestion de profils** : Permettre aux influenceurs de créer et compléter leur profil professionnel avec toutes les informations nécessaires (domaines d'expertise, plateformes sociales, tarifs)
- **Marketplace** : Offrir une interface de découverte des campagnes publicitaires avec des fonctionnalités de recherche et de filtrage
- **Gestion des candidatures** : Faciliter la candidature des influenceurs et le suivi des applications
- **Dashboard analytique** : Fournir un tableau de bord avec statistiques et indicateurs
- **Expérience utilisateur moderne** : Développer une interface utilisateur intuitive et responsive avec React et TypeScript

1.3 Technologies Utilisées

1.3.1 Backend

- **Django 5.2** : Framework web Python pour le développement backend, offrant une architecture MVC robuste et une administration intégrée

- **Django REST Framework** : Extension de Django spécialisée dans la création d'APIs REST, facilitant la sérialisation, la validation et la documentation des endpoints
- **MySQL** : Base de données relationnelle pour le stockage persistant des données avec un schéma optimisé et des relations bien définies
- **Simple JWT** : Bibliothèque pour l'implémentation de l'authentification par tokens JWT (JSON Web Tokens) avec support des tokens d'accès et de rafraîchissement

1.3.2 Frontend

- **React 18.3.1** : Bibliothèque JavaScript moderne pour la construction d'interfaces utilisateur réactives et modulaires
- **TypeScript** : Superset typé de JavaScript offrant une meilleure maintenabilité du code et une détection précoce des erreurs
- **Vite 6.3.5** : Outil de build moderne et rapide pour le développement frontend, offrant un rechargement à chaud optimisé
- **Bootstrap 5.3.3** : Framework CSS utilisé exclusivement pour la création d'interfaces responsive avec des composants pré-construits et un système de grille flexible
- **React Bootstrap 2.10.2** : Composants React basés sur Bootstrap pour faciliter l'intégration de Bootstrap dans React
- **React Router DOM 7.9.6** : Bibliothèque de routage pour la navigation entre les différentes pages de l'application React
- **Lucide React** : Bibliothèque d'icônes moderne et légère pour améliorer l'expérience visuelle
- **Recharts** : Bibliothèque de graphiques pour React permettant la visualisation de données
- **React Hook Form** : Bibliothèque pour la gestion efficace des formulaires avec validation

1.3.3 Outils et Infrastructure

- **Git** : Système de contrôle de version pour la gestion collaborative du code
- **npm** : Gestionnaire de paquets pour les dépendances JavaScript/TypeScript
- **pip** : Gestionnaire de paquets Python pour les dépendances backend
- **MySQL Connector** : Driver Python pour la connexion à la base de données MySQL

2 Architecture du Projet

2.1 Architecture Générale

Le projet suit une architecture full-stack moderne avec une séparation claire entre le backend et le frontend, communiquant via une API REST. Cette architecture assure maintenabilité, évolutivité et indépendance entre backend et frontend.

2.1.1 Architecture MVC Adaptée

Le projet implémente une architecture MVC (Model-View-Controller) adaptée au contexte full-stack :

- **Model (Backend)** : Les modèles Django dans `api/models/` représentent la structure des données et la logique métier associée. Ils définissent les entités du système (User, Influenceur, Campagne, Candidature, etc.) et leurs relations.
- **View (Frontend)** : Les pages React dans `frontend/src/pages/` constituent la couche de présentation. Elles affichent les données et gèrent l'interaction utilisateur via l'API backend.
- **Controller** :
 - **Backend** : Les vues API dans `api/views/` agissent comme contrôleurs, gérant les requêtes HTTP, la validation des données, et la logique métier avant de retourner les réponses appropriées.
 - **Frontend** : Les services dans `frontend/src/services/` gèrent la communication avec l'API, la transformation des données, et la gestion des états d'authentification.

2.2 Structure des Répertoires

La structure du projet est organisée de manière modulaire pour faciliter la maintenance et la collaboration :

2.2.1 Backend (Django)

- `backend/` : Configuration principale Django
 - `settings.py` : Configuration de l'application (base de données, middleware, CORS, JWT, etc.)
 - `urls.py` : Routes principales pointant vers l'API et l'administration
 - `wsgi.py` : Configuration WSGI pour le déploiement en production
- `api/` : Application Django principale

- `models/` : Définition des modèles de données (User, Influenceur, Campagne, etc.)
- `views/` : Vues API gérant les endpoints REST
- `serializers/` : Sérialiseurs DRF pour la transformation des données
- `urls.py` : Routes API spécifiques
- `signals.py` : Signaux Django pour automatiser certaines actions, comme la création de profils
- `admin.py` : Configuration de l'interface d'administration Django

2.2.2 Frontend (React)

- `frontend/src/` : Code source React
 - `pages/` : Pages principales de l'application (Landing, Login, Dashboard, Marketplace, etc.)
 - `components/` : Composants réutilisables (Sidebar, TopBar, formulaires, etc.)
 - `services/` : Services API et gestion de l'authentification
 - `contexts/` : Contextes React pour la gestion d'état global (AuthContext)
 - `index.css` : Fichier CSS principal, incluant Bootstrap et styles personnalisés
 - `styles/` : Fichiers CSS additionnels (styles Bootstrap personnalisés optionnels)

2.3 Communication Backend-Frontend

La communication entre le backend et le frontend s'effectue via une API REST utilisant le protocole HTTP/HTTPS. Les principales caractéristiques de cette communication sont :

- **Format JSON** : Toutes les données sont échangées au format JSON, facilitant la manipulation côté frontend
- **Authentification JWT** : Les requêtes authentifiées incluent un token JWT dans l'en-tête Authorization
- **CORS** : Configuration Cross-Origin Resource Sharing pour autoriser les requêtes depuis le frontend
- **Gestion d'erreurs** : Système unifié de gestion des erreurs avec codes HTTP appropriés et messages descriptifs

3 Modèles de Données

3.1 Architecture de la Base de Données

La base de données MySQL assure la cohérence et la fiabilité des données de la plateforme, permettant de gérer efficacement les utilisateurs, campagnes et candidatures.

3.2 Modèle User

Le modèle **User** est le modèle central du système d'authentification. Il hérite de **AbstractBaseUser** de Django, permettant une personnalisation complète du système d'authentification.

3.2.1 Relations

Le modèle **User** entretient des relations OneToOne avec les modèles **Influenceur** et **Entreprise**, permettant à chaque utilisateur d'avoir un profil spécifique selon son type.

3.3 Modèle Influenceur

Le modèle **Influenceur** représente le profil professionnel complet d'un influenceur sur la plateforme.

3.3.1 Informations de Base

- **Pseudo** : Nom d'affichage public de l'influenceur
- **Photo de profil** : URL de l'image de profil
- **Biographie** : Description personnelle et professionnelle
- **Localisation** : Localisation géographique de l'influenceur

3.3.2 Métriques et Statistiques

- **Pourcentage de complétion** : Calcul automatique basé sur les champs remplis (pseudo, bio, localisation, domaines, plateformes, tarifs)
- **Taux d'acceptation** : Pourcentage de candidatures acceptées sur le total
- **Total de candidatures** : Nombre total de candidatures soumises
- **Candidatures acceptées** : Nombre de candidatures acceptées

3.4 Modèle Entreprise et Campagne

3.4.1 Modèle Entreprise

Le modèle **Entreprise** représente les entreprises utilisant la plateforme pour publier des campagnes.

- **Nom de l'entreprise** : Nom officiel de l'entreprise
- **Description** : Présentation de l'entreprise et de ses activités
- **Secteur** : Secteur d'activité de l'entreprise
- **Taille** : Taille de l'entreprise (startup, PME, grande entreprise)

3.4.2 Modèle Campagne

Le modèle **Campagne** représente les offres publicitaires publiées par les entreprises.

- **Informations de base** : Titre, description détaillée de la campagne
- **Budget** : Budget minimum et maximum alloué à la campagne
- **Date limite** : Date limite pour les candidatures
- **Statut** : État de la campagne (brouillon, active, fermée, annulée)
- **Domaines ciblés** : Relation ManyToMany avec **DomaineExpertise** pour cibler des domaines spécifiques
- **Plateformes ciblées** : Relation ManyToMany avec **PlateformeSociale** pour cibler des plateformes spécifiques
- **Métriques** : Nombre total de candidatures reçues

3.5 Modèle Candidature

Le modèle **Candidature** représente les candidatures des influenceurs aux campagnes.

3.5.1 Caractéristiques

- **Statut** : État de la candidature (en attente, acceptée, refusée, retirée)
- **Message de motivation** : Message personnalisé de l'influenceur expliquant son intérêt
- **Prix proposé** : Tarif proposé par l'influenceur pour la collaboration
- **Date de candidature** : Timestamp automatique de la soumission
- **Contrainte d'unicité** : Un influenceur ne peut candidater qu'une seule fois par campagne

3.6 Modèle Notification

Le modèle **Notification** permet de notifier les utilisateurs des événements importants (nouvelles campagnes, réponses aux candidatures, etc.).

- **Type** : Type de notification (nouvelle campagne, réponse candidature, etc.)
- **Message** : Contenu de la notification
- **Statut de lecture** : Indicateur si la notification a été lue
- **Timestamp** : Date de création de la notification

4 Système d'Authentification

4.1 Architecture JWT

Le projet utilise JSON Web Tokens (JWT) pour gérer les sessions utilisateurs de manière sécurisée et stateless. Cette approche permet d'authentifier les requêtes sans stocker de session côté serveur, tout en garantissant la sécurité des échanges et la compatibilité avec différents clients (web, mobile).

4.2 Processus d'Inscription et de Connexion

Inscription : L'utilisateur fournit un email, un mot de passe et son type (influenceur ou entreprise). Le système vérifie les données, crée le compte avec un mot de passe hashé, génère les tokens JWT (access et refresh) et crée automatiquement le profil associé.

Connexion : L'utilisateur saisit son email et son mot de passe. Après validation, les tokens JWT sont générés et renvoyés au client pour authentifier les futures requêtes. La dernière connexion est également enregistrée. Le système utilise l'email comme identifiant grâce à un serializer personnalisé.

4.3 Gestion et Sécurité des Tokens

Deux types de tokens sont utilisés :

- **Access Token** : court terme, pour authentifier les requêtes API
- **Refresh Token** : long terme, pour obtenir de nouveaux tokens d'accès

Les tokens sont sécurisés grâce à leur expiration automatique et à la rotation des refresh tokens. Les tokens expirés sont invalidés pour empêcher toute réutilisation abusive.

4.4 Protection des Routes

Les endpoints backend sont protégés par des permissions Django REST Framework, garantissant que seules les requêtes authentifiées accèdent aux ressources sensibles. Côté frontend, les tokens sont stockés et injectés automatiquement dans les requêtes API, avec redirection vers la page de connexion si l'utilisateur n'est pas authentifié ou si le token a expiré.

5 Gestion des Profils

5.1 Architecture de Gestion des Profils

Le système de gestion des profils permet aux influenceurs de créer, consulter et mettre à jour leur profil professionnel. Ce profil centralise l'ensemble des informations nécessaires pour participer aux campagnes proposées sur la plateforme.

5.2 Profil Influenceur

Le profil influenceur est structuré autour de plusieurs éléments essentiels permettant de présenter clairement l'utilisateur aux entreprises.

5.2.1 Composants du Profil

Le profil influenceur comprend :

- **Informations de base** : pseudo, biographie, localisation et photo de profil
- **Domaines d'expertise** : sélection d'un ou plusieurs domaines d'activité
- **Plateformes sociales** : plateformes utilisées avec indication du nombre d'abonnés
- **Tarifs** : prix proposés selon le type de contenu (post, story, vidéo)

5.2.2 Complétion du Profil

Un pourcentage de complétion du profil est calculé automatiquement en fonction des informations renseignées par l'influenceur. Cet indicateur permet d'encourager les utilisateurs à compléter leur profil afin d'optimiser leur visibilité auprès des entreprises.

5.3 Mise à Jour du Profil

Les influenceurs peuvent modifier à tout moment les informations de leur profil via l'interface de la plateforme. Les modifications prennent en compte l'ensemble

des sections du profil (informations personnelles, domaines, plateformes et tarifs) et sont enregistrées de manière cohérente.

5.4 Récupération des Informations Utilisateur

La plateforme permet de récupérer les informations de l'utilisateur connecté afin d'afficher dynamiquement son profil et son niveau de complétion. Ces données sont structurées de manière à être directement exploitables par le frontend pour l'affichage des profils et des tableaux de bord.

6 Interface Utilisateur Frontend

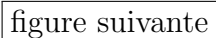
6.1 Architecture Frontend

L'interface utilisateur de la plateforme Influmatch est développée avec React et TypeScript. Ce choix technologique permet de proposer une application moderne, dynamique et offrant une bonne expérience utilisateur.

6.2 Pages Principales

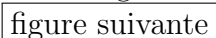

6.2.1 Page d'Accueil

La page d'accueil (`LandingPage.tsx`) constitue le point d'entrée principal de la plateforme. Elle présente le concept du service, les fonctionnalités principales ainsi que des appels à l'action invitant les visiteurs à s'inscrire ou à se connecter.

La  illustre la page d'accueil de la plateforme Influmatch.

6.2.2 Pages d'Authentification

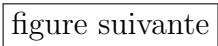
Les pages d'inscription (`SignupPage.tsx`) et de connexion (`LoginPage.tsx`) permettent aux utilisateurs de créer un compte ou d'accéder à leur espace personnel. Elles intègrent des formulaires avec validation des données et gestion des erreurs afin de guider l'utilisateur tout au long du processus.

La  illustre la page d'inscription et  illustre la page de connexion de la plateforme Influmatch.

6.2.3 Dashboard

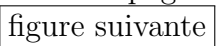
Le dashboard (`DashboardPage.tsx`) représente l'espace principal des utilisateurs authentifiés. Il offre une vue d'ensemble des informations importantes, telles

que le niveau de complétion du profil, les statistiques personnelles et un accès rapide aux fonctionnalités clés de la plateforme.

La  présente le tableau de bord utilisateur après authentification.

6.2.4 Marketplace

La marketplace (`MarketplacePage.tsx`) permet aux influenceurs de découvrir les campagnes publiées par les entreprises. Les utilisateurs peuvent consulter les offres disponibles, effectuer des recherches et appliquer différents filtres afin de trouver des campagnes correspondant à leurs critères.

La  illustre la marketplace permettant la recherche et la consultation des campagnes.

6.2.5 Page de Détails d’Offre

La page de détails d’une offre (`OfferDetailPage.tsx`) affiche l’ensemble des informations liées à une campagne, incluant sa description, son budget et les critères recherchés. Elle permet également à l’influenceur de postuler directement à la campagne (actuellement non fonctionnel, données fictives).

La page de détails d’une offre de campagne se trouve sur la  que la marketplace.

6.2.6 Gestion des Candidatures

La page de gestion des candidatures (`ApplicationsPage.tsx`) permet aux influenceurs de suivre l’ensemble de leurs candidatures et d’en consulter le statut. Cette interface facilite le suivi des réponses des entreprises.

La  illustre les différentes candidatures que les influenceurs ont pu envoyer.

6.2.7 Profil Utilisateur

La page profil (`ProfilePage.tsx`) permet aux utilisateurs de consulter et de modifier leurs informations personnelles et professionnelles. Elle constitue un élément central de la plateforme, notamment pour les influenceurs souhaitant valoriser leur profil auprès des entreprises.

La  présente l’interface de gestion du profil de l’influenceur.

6.2.8 Complétion du Profil

Une interface dédiée (`ProfileCompletionPage.tsx`) guide les influenceurs dans la complétion progressive de leur profil. Un indicateur de progression permet de visualiser le niveau de complétion et d'encourager l'utilisateur à renseigner l'ensemble des informations requises.

La page de complétion du profil se trouve sur la même figure que la page du profil.

6.3 Composants Réutilisables

La plateforme utilise des composants réutilisables pour le layout et les formulaires, assurant la cohérence et l'ergonomie de l'interface.

6.4 Gestion d'État

La gestion de l'état global est assurée via React Context API, tandis que chaque composant conserve son état local pour les données spécifiques à la page.

6.5 Routing et Navigation

Le routing est géré avec React Router DOM, permettant de définir des routes publiques et protégées ainsi que des redirections automatiques selon l'état d'authentification.

6.6 Service API

6.6.1 Architecture du Service

Le service API (`services/api.ts`) centralise toute la communication avec le backend :

- **Configuration centralisée** : URL de base de l'API configurée via variable d'environnement
- **Gestion des tokens** : Fonctions pour stocker, récupérer et supprimer les tokens JWT
- **Requêtes HTTP** : Fonctions génériques pour effectuer des requêtes avec injection automatique des tokens
- **Gestion d'erreurs** : Traitement centralisé des erreurs avec messages utilisateur appropriés

6.6.2 Endpoints API

Le service expose des fonctions pour chaque endpoint :

- **authAPI.login** : Connexion et récupération des tokens
- **authAPI.signup** : Inscription et génération automatique des tokens
- **authAPI.getCurrentUser** : Récupération des informations utilisateur
- **authAPI.logout** : Déconnexion et nettoyage des tokens
- **profileAPI.updateProfile** : Mise à jour du profil influenceur

7 Endpoints API

7.1 Architecture REST

L'API suit les principes REST (Representational State Transfer) avec des URLs sémantiques, des méthodes HTTP adaptées, des codes de statut standard et un format JSON pour toutes les réponses.

7.2 Endpoints d'Authentification

Les principaux endpoints d'authentification incluent :

- **POST /api/auth/register/** : Inscription d'un nouvel utilisateur
- **POST /api/auth/token/** : Connexion et obtention des tokens JWT
- **POST /api/auth/token/refresh/** : Rafraîchissement du token d'accès
- **GET /api/auth/user/** : Récupération des informations de l'utilisateur connecté

7.3 Endpoints de Profil

- **POST/PUT /api/profile/update/** : Mise à jour du profil influenceur, incluant informations personnelles, domaines, plateformes et tarifs

7.4 Gestion des Erreurs

L'API s'appuie sur les codes HTTP standard (200, 201, 400, 401, 403, 404, 500) pour indiquer le résultat des requêtes. Toutes les erreurs sont renvoyées dans un format JSON unifié contenant un message clair destiné au frontend, avec des détails techniques fournis par Django REST Framework lorsque nécessaire.

8 Fonctionnalités Techniques

8.1 Système de Signaux Django

Django Signals est utilisé pour automatiser certaines actions lors d'événements spécifiques. Par exemple, lors de la création d'un utilisateur, un signal déclenche automatiquement la création du profil associé (influenceur ou entreprise), en vérifiant que le profil n'existe pas déjà. Cette automatisation garantit l'intégrité des données et évite les profils manquants.

8.2 Interface d'Administration Django

L'interface d'administration offre une gestion complète des données pour les administrateurs. Les modèles principaux (User, Influenceur, Entreprise, Campagne, Candidature, Notifications) sont enregistrés avec des configurations personnalisées permettant :

- Affichage des champs clés dans les listes
- Filtres et recherche textuelle pour faciliter la navigation
- Protection des champs automatiques en lecture seule
- Actions groupées sur plusieurs objets

8.3 Configuration CORS

Le module `django-cors-headers` est utilisé pour autoriser les requêtes provenant du frontend React. Seuls les domaines approuvés peuvent communiquer avec l'API, sécurisant ainsi les échanges entre frontend et backend.

8.4 Configuration de la Base de Données

8.4.1 MySQL Connector

La connexion à MySQL utilise MySQL Connector pour Django, offrant :

- Configuration via variables d'environnement pour plus de sécurité
- Gestion automatique du pool de connexions
- Support des transactions pour garantir l'intégrité des données

8.4.2 Migrations Django

Le système de migrations gère l'évolution du schéma de base de données :

- Génération automatique des migrations lors de modifications de modèles
- Versioning des migrations avec possibilité d'application ou d'annulation
- Historique complet des modifications du schéma

8.5 Sécurité

8.5.1 Protection CSRF

Django fournit une protection CSRF intégrée pour toutes les requêtes modifiant des données (POST, PUT, DELETE).

8.5.2 Validation des Données

Toutes les données sont validées côté serveur, avec une validation supplémentaire côté frontend pour améliorer l'expérience utilisateur. Les messages d'erreur sont clairs et descriptifs.

8.5.3 Hashage des Mots de Passe

Les mots de passe sont sécurisés via PBKDF2 avec SHA256, offrant une protection robuste contre les attaques par force brute.

9 Base de Données

9.1 Schéma Relationnel

La base de données MySQL est organisée selon un schéma relationnel optimisé pour la performance et l'intégrité des données.

9.2 Tables Principales

9.2.1 Utilisateurs

Table centrale stockant tous les utilisateurs (influenceurs et entreprises) :

- Clé primaire : ID auto-incrémenté
- Index : email (unique) et type_utilisateur
- Contraintes : email unique et type d'utilisateur valide

9.2.2 Influenceurs

- Clé étrangère : OneToOne vers utilisateurs
- Index : `pourcentage_completion_profil`, `localisation`
- Métriques : statistiques comme taux d'acceptation et total candidatures

9.2.3 Entreprises

- Clé étrangère : OneToOne vers utilisateurs
- Informations : `nom`, `description`, `secteur`, `taille`

9.2.4 Campagnes

- Clé étrangère : ForeignKey vers entreprises
- Statut : `brouillon`, `active`, `fermée`, `annulée`
- Budget : `minimum` et `maximum`
- Dates : `date limite de candidature`

9.2.5 Candidatures

- Clés étrangères : ForeignKey vers campagnes et influenceurs
- Contrainte : un influenceur ne peut candidater qu'une fois par campagne
- Statut et timestamp : suivi du statut et date de candidature

9.3 Tables de Liaison

9.3.1 Domaines d'Expertise

- `domaines_expertise` : liste des domaines
- `influenceur_expertise` : liaison ManyToMany influenceurs domaines
- `campagne_domaine` : liaison ManyToMany campagnes domaines

9.3.2 Plateformes Sociales

- `plateformes_sociales` : liste des plateformes
- `influenceur_plateformes` : liaison avec métadonnées (nombre d'abonnés)
- `campagne_plateforme` : liaison ManyToMany campagnes plateformes

9.4 Relations et Intégrité

9.4.1 Types de Relations

- OneToOne : User Influenceur, User Entreprise
- OneToMany : Entreprise → Campagne, Influenceur → Candidature, Campagne → Candidature
- ManyToMany : Influenceur DomaineExpertise, Campagne DomaineExpertise, Influenceur PlateformeSociale, Campagne PlateformeSociale

9.4.2 Contraintes d'Intégrité

- CASCADE : suppression en cascade pour maintenir l'intégrité
- UNIQUE : éviter les doublons (email, candidature unique par campagne)
- NOT NULL : champs obligatoires pour garantir la complétude

9.5 Optimisations

9.5.1 Index

- email, type_utilisateur, pourcentage_completion_profil, localisation

9.5.2 Requêtes Optimisées

- Select Related : éviter les requêtes N+1 pour ForeignKey
- Prefetch Related : optimiser les relations ManyToMany
- Agrégation : calcul des statistiques directement en base

10 Configuration et Déploiement

Le projet utilise Django pour le backend et React/Vite pour le frontend. Toutes les informations de configuration (variables d'environnement, installation, lancement) sont documentées dans le fichier README du projet.

11 Conclusion

Le projet **InfluMatch** a permis de développer une plateforme complète de mise en relation entre influenceurs et entreprises. Les principales fonctionnalités implémentées comprennent un système d'authentification sécurisé avec JWT, la

gestion complète des profils influenceurs et entreprises, un espace marketplace pour découvrir les offres, un suivi des candidatures avec différents statuts, ainsi qu'un tableau de bord analytique personnalisé. L'expérience utilisateur est moderne et réactive grâce à l'utilisation de React avec TypeScript, garantissant une interface intuitive et sécurisée.

11.1 Améliorations Futures

Plusieurs axes d'amélioration pourraient enrichir la plateforme et offrir davantage de fonctionnalités aux utilisateurs :

- Système de messagerie en temps réel entre influenceurs et entreprises
- Intégration d'un système de paiement pour faciliter les transactions
- Tableaux de bord analytiques avancés pour suivre la performance des campagnes
- Développement d'une application mobile ou Progressive Web App
- Recommandations d'influenceurs via un algorithme intelligent et intégration avec les APIs sociales
- Notifications en temps réel et support multilingue

11.2 Apprentissages et Défis

Le développement de ce projet a été une expérience riche en apprentissages. Nous avons renforcé notre compréhension de l'architecture full-stack, de la communication entre frontend et backend, de la modélisation de bases de données relationnelles et de la conception d'API REST sécurisées. L'utilisation de TypeScript et de React pour le frontend nous a permis d'améliorer la maintenabilité et la robustesse du code. Nous aurions aimé pouvoir finaliser complètement le projet, afin de démontrer pleinement l'étendue de ce que nous sommes capables de réaliser. Même si nous avons eu recours à l'assistance d'outils d'intelligence artificielle, chaque ligne de code a été comprise, adaptée et intégrée avec discernement. Cela a renforcé notre compréhension du développement web, et rendu l'apprentissage d'autant plus motivant et concret.

En somme, InfluMatch constitue un projet ambitieux qui nous a permis de mettre en pratique nos compétences techniques tout en préparant le terrain pour de futures améliorations et fonctionnalités avancées.

12 Webographie

Références

[ChatGPT] <https://chatgpt.com/?oai-dm=1>

[Claude.ai] <https://claude.ai>

13 Annexes

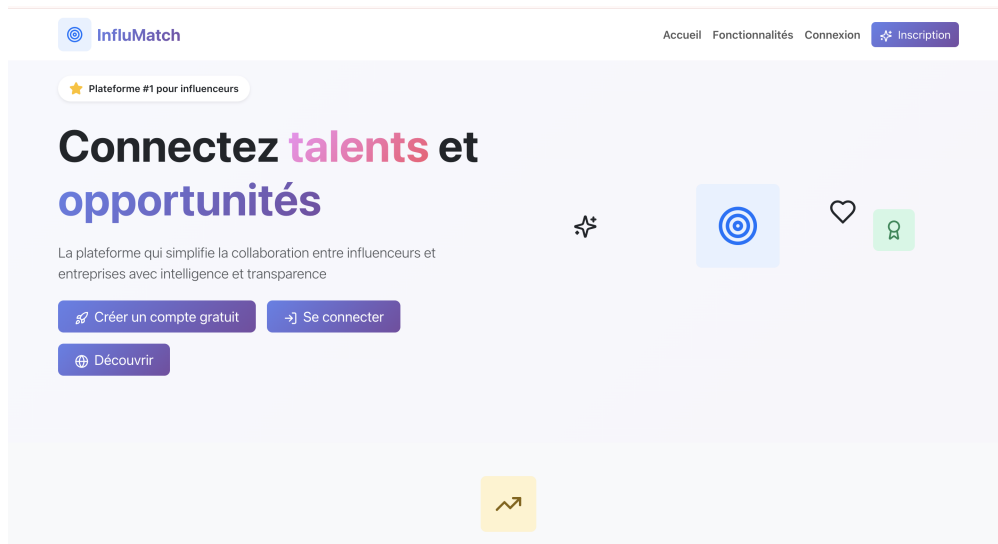


FIGURE 1 – Page d'accueil.



Créer un compte

Rejoignez InfluMatch en quelques secondes

Type de compte

Influenceur ✓
Trouvez des opportunités

Entreprise
Bientôt disponible

Email

votre@email.com

Mot de passe

Minimum 8 caractères

Confirmer le mot de passe

Confirmer le mot de passe

Créer mon compte

Vous avez déjà un compte ? [Se connecter](#)

FIGURE 2 – Page d'inscription.



Bon retour !

Connectez-vous à votre compte InfluMatch

Email

Mot de passe

☐ Se souvenir de moi

Se connecter

[Mot de passe oublié ?](#)

Pas encore de compte ? [S'inscrire](#)

FIGURE 3 – Page de connexion.

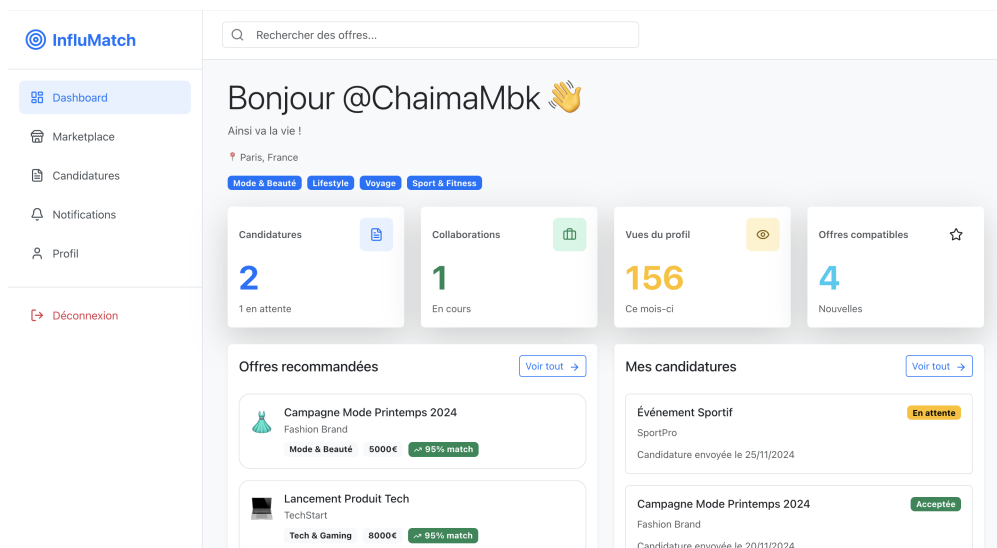


FIGURE 4 – Dashboard.

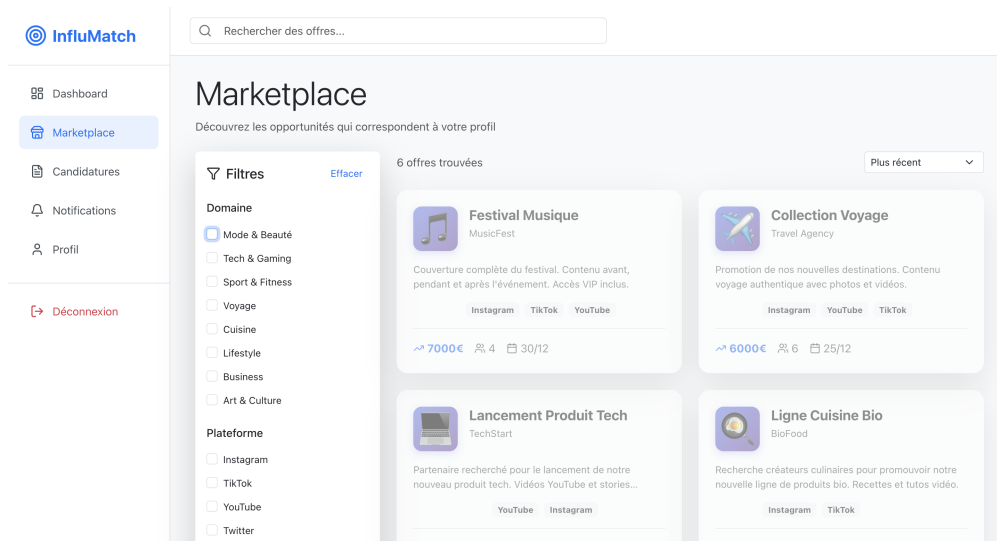


FIGURE 5 – Marketplace et page de détails d'une offre de campagne.

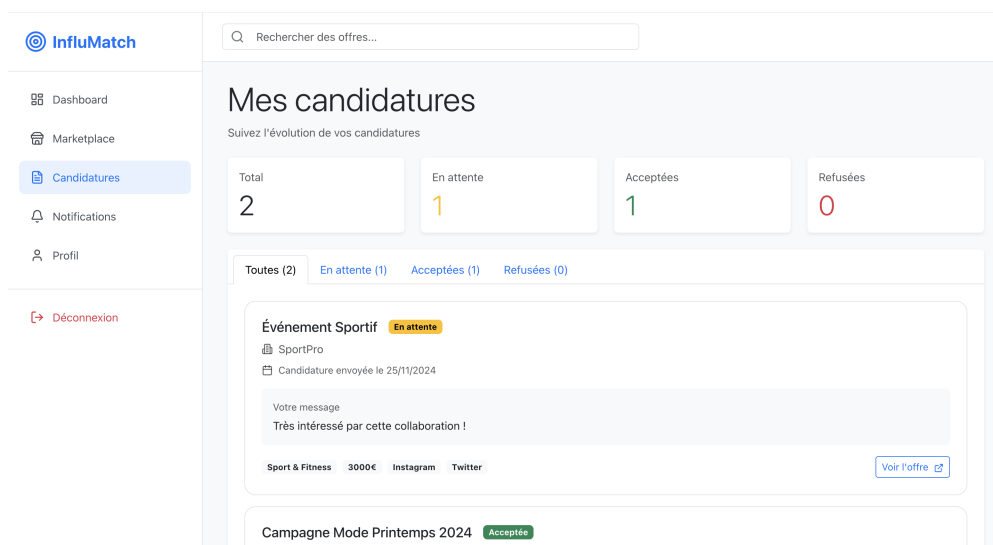


FIGURE 6 – Candidatures.

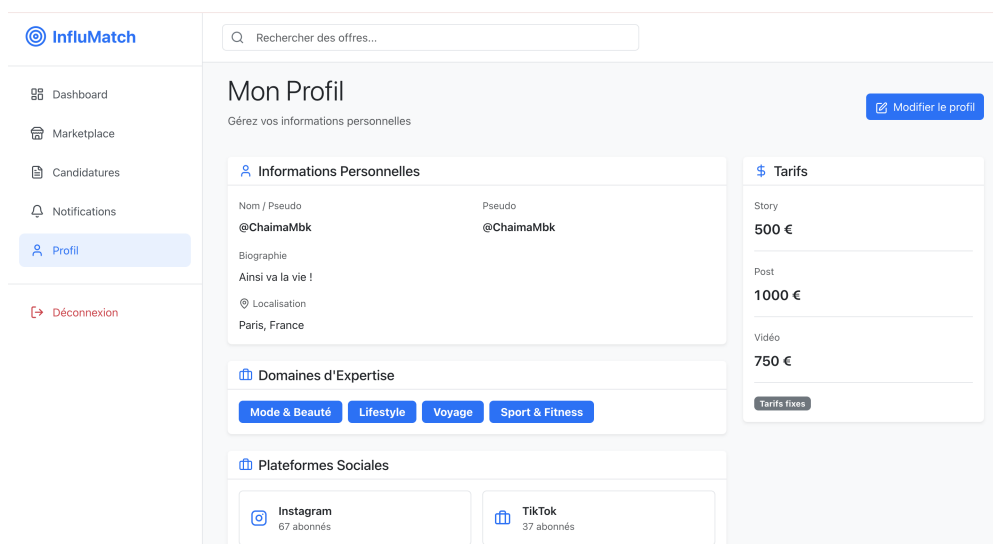


FIGURE 7 – Profil utilisateur.

Fin du rapport