

Cahier des charges : AdopteUnDev

1. Objectif général

Créer une application web permettant aux développeurs et aux entreprises de créer des profils ou des fiches de postes et de les faire correspondre selon un modèle inspiré des sites de rencontres. L'objectif est d'optimiser la mise en relation entre développeurs et recruteurs via des fonctionnalités conviviales et innovantes.

2. Fonctionnalités proposées

2.1. Gestion des Utilisateurs

- **Inscription distincte** : développeur ou entreprise.
- **Profil public/privé** : option pour masquer les informations sensibles (contact, salaire).
- **Rôles** : gestion des droits utilisateur via `ROLE_DEV` et `ROLE_COMPANY`.

2.2. Création et Gestion des Profils/Fiches de postes

Pour les développeurs :

- Profil détaillé avec :
 - Nom et prénom.
 - Localisation.
 - Langages de programmation (choix dans une liste).
 - Niveau d'expérience (notation sur 5).
 - Salaire minimum souhaité.
 - Courte biographie.
 - Avatar (upload via Symfony Form).

Pour les entreprises :

- Création de fiches de postes avec :
 - Titre du poste.
 - Localisation.
 - Technologies recherchées.
 - Niveau d'expérience requis.
 - Salaire proposé.
 - Description détaillée.

2.3. Système de Matching (Correspondance)

- Suggestions dynamiques de profils ou de postes selon des critères :
 - Langages ou technologies.
 - Salaire (fourchette souhaitée/proposée).
 - Localisation géographique.
 - Niveau d'expérience.

2.4. Pages d'Accueil Dynamiques

- **Pour les entreprises** :
 - Affichage des **Profils de développeurs les plus consultés**.
 - Affichage des **3 derniers profils créés**.
- **Pour les développeurs** :
 - Affichage des **Postes les plus populaires**.
 - Affichage des **3 dernières offres publiées**.

2.5. Recherche Avancée

- Système de filtres :
 - Développeurs ou postes.
 - Langages/technologies.
 - Localisation.
 - Salaire souhaité/proposé.
 - Niveau d'expérience.

2.6. Évaluations et Feedbacks

- Développeurs :
 - Possibilité d'évaluer d'autres développeurs sur leur **niveau d'expérience** (notation sur 5).
 - Affichage d'une moyenne des évaluations.

2.7. Fonctionnalités sociales

- **Favoris** : ajouter un développeur ou **une fiche de poste à une liste personnelle**.
- **Messagerie intégrée** (optionnelle) : communication directe entre développeurs et entreprises.
- **Tableau de bord personnalisé** :
 - Notifications des nouvelles correspondances.
 - Historique des recherches ou contacts.

2.8. Suivi et Analyses

- **Statistiques utilisateur** :
 - Pour les développeurs : nombre de vues de leur profil.
 - Pour les entreprises : nombre de vues des fiches de poste.
- **Top profils/postes** : section affichant les plus demandés.

2.9. Backup et Déploiement

- Fonctionnalité de **sauvegarde** pour exporter les données en cas de besoin.
- Déploiement via Docker pour un environnement standardisé.

3. Contraintes techniques

Backend et Frontend

- Backend : Symfony pour l'intégralité des fonctionnalités serveur.
- Frontend : rendu principal via Twig (HTML généré côté serveur). Minimalisme dans l'utilisation de JavaScript.

Git et Collaboration

- Utilisation obligatoire de **Git** pour le suivi du projet.
- Contributions régulières par **tous les membres** du groupe.
- Historique Git analysé pour évaluer la collaboration et l'implication.

Livrables obligatoires

- **Code complet** : hébergé sur un dépôt Git public/privé (si privé, mon mail virginie.sans@irisa.fr pour me mettre dedans) et dans le readme, le nom des étudiants qui composent le groupe.
- **Vidéo de démonstration** des fonctionnalités.
- **Fichier Docker** fonctionnel pour la mise en production.

Sécurité

- Chiffrement des mots de passe.
- Protection contre les failles XSS et CSRF.

4. Réalisation et Évaluation

Groupes

- Projet réalisé en **groupes de 3 ou 4 étudiants** (à composer à votre convenance)
- Travail collaboratif obligatoire avec répartition des tâches.

Date de rendu

- Début janvier (date précise à définir).

Critères d'évaluation

1. Respect des fonctionnalités demandées.
2. Qualité du code (normes PSR, bonnes pratiques Symfony).
3. Contribution régulière et équitable via Git.
4. Fonctionnalités avancées ou innovantes supplémentaires.
5. Qualité de la démonstration vidéo.