

Rapport de projet « Job Finder »

Auteurs : Rosa Bouzid & Céline Kerrar

1. Contexte et objectif :

Job Finder est un site d'offres d'emploi entièrement dynamique dédiée à la mise en relation entre candidat et recruteurs. Ce projet répond à deux besoins principaux :

Pour les candidats:

la consultation des offres d'emploi (CDI, CDD, Stage, Alternance, Freelance) , la possibilité de recherche et de postuler aux annonces et de visualiser détaillées des offres mais aussi de toutes ses candidatures.

Pour les recruteurs :

la publication de nouvelles offres d'emplois , la modification et suppression des ses propres publications et la consultation des candidatures reçus.

choix techniques :

- **Frontend** : HTML, CSS, JavaScript .
- **Backend** : PHP, stockage CSV, gestion de sessions.
- **Organisation** : réutilisation de modèles vus en TP.

2. Architecture générale

```
/
| index.php      ← page d'accueil
| offres.php     ← liste et filtrage d'offres
| offre.php      ← détail d'une offre et le bouton Postuler
| postuler.php   ← formulaire de candidature
| inscription.php← création de compte (candidat/recruteur)
| connexion.php  ← authentification
| dashboard-candidat.php ← tableau de bord du candidat
| dashboard-recruteur.php ← tableau de bord du recruteur
```

```
├─ ajouter-offre.php,  
├─ modifier-offre.php,  
├─ supprimer-offre.php  
├─ mes-candidatures.php  
├─ voir-candidatures.php,  
├─ profil.php      ← espace du candidat  
├─ deconnexion.php  
├─ css/  
|   └─ style.css  
├─ js/  
|   └─ script.js  
└─ data/  
    ├─ utilisateurs.csv      ← comptes et rôles  
    ├─ offres.csv           ← base d'offres  
    ├─ candidatures.csv     ← candidatures envoyées  
    ├─ profils.csv          ← données de profil  
    ├─ job_recherche.csv    ← préférences de poste recherchées  
    ├─ competences.csv      ← liste de compétences par candidat  
    ├─ experiences.csv  
    └─ diplomes.csv
```

3. Structure et contenu des fichiers CSV

on a utilisé un système de stockage organisé en fichier CSV pour gérer les utilisateurs, offres d'emploi, candidatures et profils :

utilisateurs.csv	id, role(recruteur ou candidat) , login, nom, email, hashmdp
offres.csv	id de offre, id du recruteur, titre, entreprise, description, contrat(CDI , stage ...), lieu, salaire min, salaire max, date_debut, date de publication ,
candidatures.csv	id_offre, id du candidat, id du recruteur, date_cand, chemin vers le cv, chemin vers la lettre de motivation

profils.csv	id du candidat, chemin vers la photo, adresse, etudes, experiences , compétences, public_cv
job_recherche.csv	id du candidat, job, job, job_lieu, salaire_min, salaire_max, cv_lien
competences.csv	id du candidat, compétence
experiences.csv	id de l'expérience , id du candidat, poste, entreprise, lieu, début, fin, description
diplomes.csv	id du diplome, id du candidat, nom, etablissement, niveau, date, obtenu(oui/non), description

4. Fonctionnalités détaillées

4.1. Candidat

1. Page d'accueil ([index.php](#))

- Hero slogan “Notre job, vous aider à choisir le vôtre”
- Barre de recherche “QUOI ? / OÙ ?” + 5 boutons de filtre rapide
- Sélection de rôle (→ [inscription.php?role=...](#))

2. Liste d'offres ([offres.php](#))

3.

- Lecture de [offres.csv](#), filtre serveur sur GET : [titre](#), [lieu](#), [contrat](#), tri ([date/salaire](#))

4. Détail d'offre ([offre.php](#))

- Affichage des détails, bouton “Postuler” si rôle=Candidat connecté.

5. Postuler ([postuler.php](#))

- Vérification session & rôle
- Formulaire multipart : [cv](#) (PDF), [lm](#) (optionnel ou obligatoire)

- Gestion de la duplication (vérifier dans `candidatures.csv`)
 - Enregistrement dans `candidatures.csv` et déplacement des fichiers vers `/uploads`
6. **Dashboard candidat** (`dashboard-candidat.php`)
- Listing des offres valides
 - Accès aux **Mes candidatures**, **Mon profil**, **Déconnexion**
7. **Profil candidat** (`profil.php`)
- Mise à jour d'**informations personnelles** et photo
 - Définition du **job recherché** (titre, contrat, lieu, salaire, CV)
 - Gestion CRUD des **compétences**, **expériences**, **diplômes**
 - Stockage dans les CSV respectifs
8. **Mes candidatures** (`mes-candidatures.php`)
- Affiche toutes les lignes de `candidatures.csv` pour `id_candidat`

4.2. Recruteur

1. **Inscription / Connexion** (`inscription.php` / `connexion.php`)
- Unicité des emails (candidat ≠ recruteur)
 - Hashage sécurisé (bcrypt)
2. **Dashboard recruteur** (`dashbord-recruteur.php`)
- Listing des offres publiées
 - Boutons "Ajouter", "Modifier", "Supprimer"
3. **Ajouter / Modifier / Supprimer offre**
- Lecture et écriture directes dans `offres.csv`
 - Permissions un recruteur ne peut modifier/supprimer que **ses** OFFRES

4. Voir candidatures reçues ([voir-candidatures.php](#))

- Jointure [candidatures.csv](#) et [utilisateurs.csv](#) pour détails candidats
- Téléchargement direct des CV

5. Voir tous les profils ([voir-candidats.php](#))

- Filtre sur [profils.csv](#) (public_cv =oui), affichage sommaire

5. Scénarios d'usage

1. **Katia une candidate s'inscrit** recherche “analyste” + “Paris” + “CDD”, clique sur une offre, télécharge son CV et sa Lettre de motivation , reçoit la confirmation et la voit apparaître dans “ Mes candidatures”
2. **Nima un recruteur s'inscrit**, crée une offre CDI pour par exemple “Développeur PHP”, consulte les candidatures arrivées en temps réel.
3. **Katia revient plus tard, peut** modifie son profil, ajoute de nouvelles compétences et actualise son CV, puis postule à une seconde offre.

6. Conclusion

arrivée a terme de JobFinder on a développé cette plateforme dynamique répondant aux besoins des candidats et des recruteurs, leurs permettant de consulter et postuler a des offre d'emploie tandis que de l'autre coté de publier et gérer offres et leurs candidatures en séparer clairement le backend et le frontend , implémentant des fonctionnalités en js comme le filtrage des offres et le PHP a permis de genrer efficacement les data via des fichiers CSV tout en les assurant une securité speciale avec un systeme de session pour différencier les 2 utilisateurs .