

# Rapport de Projet BDR

Cseres Leonard, Laydu Aude, Gerber Tristan

26 janvier 2025

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Cahier des charges</b>	<b>3</b>
2.1	Objectif de l'application . . . . .	3
2.2	Fonctionnalités principales . . . . .	3
<b>3</b>	<b>Conception</b>	<b>3</b>
3.1	Entités Principales . . . . .	3
3.2	Modélisation Conceptuelle (Schéma EA) . . . . .	4
3.3	Modélisation Relationnelle . . . . .	4
3.4	Contraintes et Intégrités Référentielles . . . . .	5
3.4.1	Stratégies ON DELETE et ON UPDATE . . . . .	5
3.4.2	Triggers de Validation . . . . .	8
3.4.3	Contraintes Métier . . . . .	9
<b>4</b>	<b>Application</b>	<b>11</b>
4.1	Technologies Utilisées . . . . .	11
4.2	Structure de l'Application . . . . .	11
4.3	Manuel Utilisateur . . . . .	11
4.3.1	Introduction . . . . .	11
4.3.2	Création d'une Offre . . . . .	11
4.3.3	Liste des Offres . . . . .	12
4.3.4	Détail d'une Offre . . . . .	12
4.3.5	Modification d'une Offre . . . . .	12
4.3.6	Filtre des Offres . . . . .	14
4.3.7	Création d'un Candidat . . . . .	15
4.3.8	Liste des Candidats . . . . .	15
4.3.9	Détail d'un Candidat . . . . .	15
4.3.10	Modification d'un Candidat . . . . .	17
4.3.11	Filtre des Candidats . . . . .	17
<b>5</b>	<b>Conclusion</b>	<b>20</b>
<b>A</b>	<b>Annexes</b>	<b>21</b>
A.1	Guide d'Installation et d'Utilisation . . . . .	21

A.1.1 Avec Docker . . . . .	21
A.1.2 Sans Docker . . . . .	21

## 1 Introduction

Il est parfois compliqué pour les ressources humaines de suivre tous les candidats et offres d'emploi. C'est la problématique que nous avons choisi pour le projet de BDR.

Dans le cadre de ce projet, nous devons :

1. Écrire le cahier des charges
2. Modéliser un schéma EA conceptuel de la base de données
3. Transformer le schéma EA en schéma relationnel et l'implémenter avec Postgresql
4. Réaliser des requêtes, vues et triggers / procédures stockées
5. Coder une application et la lier à la base de données via une API

## 2 Cahier des charges

Consulter le cahier des charges au format PDF

### 2.1 Objectif de l'application

Développer une application pour permettre aux recruteurs de gérer efficacement les processus de recrutement, incluant la gestion des candidats, des postes et des entretiens.

### 2.2 Fonctionnalités principales

- Gestion des candidats: ajout, modification, suppression, suivi des interactions, statut.
- Gestion des offres d'emploi: création, gestion des offres, suivi des candidatures.
- Gestion des entretiens: planification et suivi des retours.
- Suivi du processus de recrutement: vue d'ensemble des candidats par poste.
- Authentification de base pour les recruteurs.

## 3 Conception

### 3.1 Entités Principales

- **Personne**: Entité racine représentant tous les acteurs du système
- **Candidat**: Spécialisation de Personne, représentant les candidats
- **Recruteur**: Spécialisation de Personne, représentant les recruteurs
- **Interaction**: Modélise les différents types de communications
- **Offre**: Représente les opportunités d'emploi
- **Domaine**: Catégorise les compétences et secteurs d'activité

### 3.2 Modélisation Conceptuelle (Schéma EA)

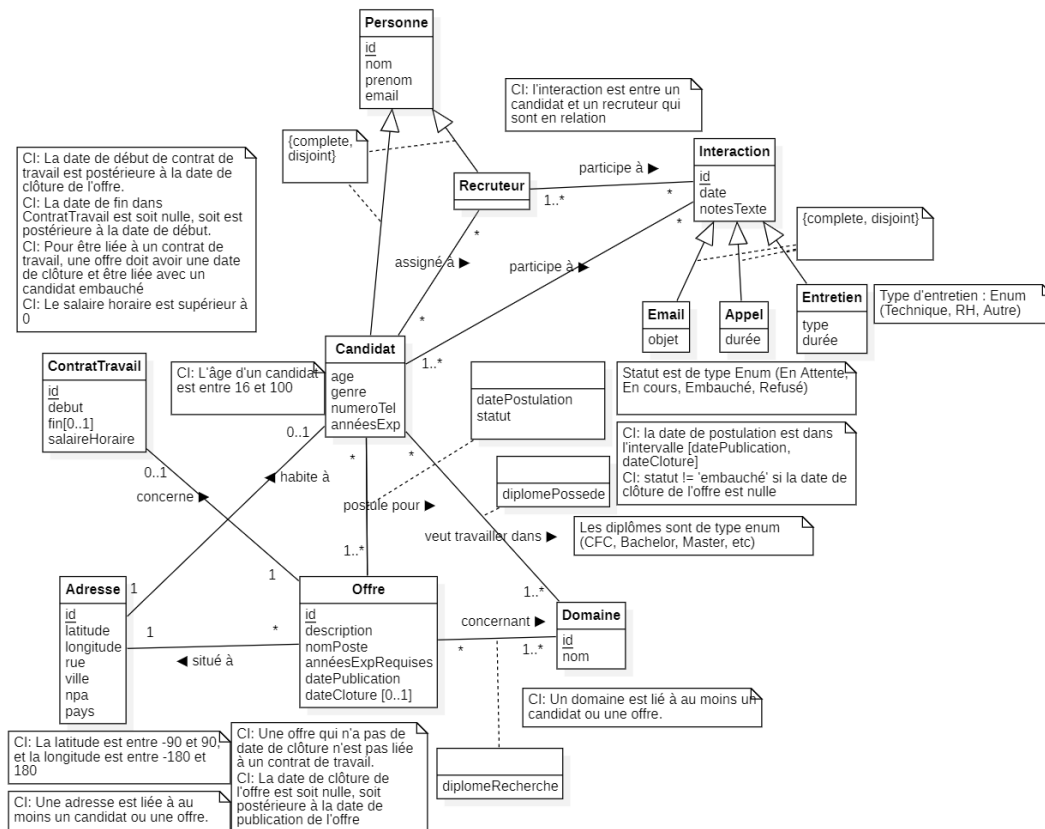


Figure 1: Schema EA

### 3.3 Modélisation Relationnelle

Le modèle relationnel permet de transformer les informations du schéma EA réalisé précédemment en un langage plus proche du code, sans pour autant en perdre la simplicité. Voici la modélisation que nous avons mise en place :

Personne(id, nom, prenom, email)

Recruteur(idPersonne)

Recruteur.idPersonne reference Personne.id

Adresse(id, latitude, longitude, rue, ville, npa, pays)

Candidat(idPersonne, age, genre, numeroTel, annéesExp, idAdresse)

Candidat.idPersonne reference Personne.id

Candidat.idAdresse reference Adresse.id Candidat.idAdresse NOT NULL UNIQUE

Recruteur\_Candidat(idRecruteur, idCandidat)

Recruteur\_Candidat.idRecruteur reference Recruteur.idPersonne

Recruteur\_Candidat(idCandidat reference Candidat.idPersonne  
 Interaction(id, date, notesTexte)  
 Recruteur\_Interaction(idRecruteur, idInteraction)  
 Recruteur\_Interaction.idRecruteur reference Recruteur.idPersonne  
 Recruteur\_Interaction.idInteraction reference Interaction.id  
 Candidat\_Interaction(idCandidat, idInteraction)  
 Candidat\_Interaction.idCandidat reference Candidat.idPersonne  
 Candidat\_Interaction.idInteraction reference Interaction.id  
 Email(idInteraction, objet)  
 Email.idInteraction reference Interaction.id  
 Appel(idInteraction, duree)  
 Appel.idInteraction reference Interaction.id  
 Entretien(idInteraction, typeEntretien, duree)  
 Entretien.idInteraction reference Interaction.id  
 Offre(id, descriptionOffre, nomPoste, annneesExpRequises, datePublication, dateCloture, idAdresse)  
 Offre.idAdresse reference Adresse.id  
 Offre.idAdresse NOT NULL UNIQUE  
 Contrat\_Travail(id, debut, fin, salaireHoraire, idOffre)  
 Contrat\_Travail.idOffre reference Offre.id  
 Contrat\_Travail.idOffre NOT NULL, UNIQUE  
 Candidat\_Offre(idCandidat, idOffre, datePostulation, statut)  
 Candidat\_Offre.idCandidat reference Candidat.idPersonne  
 Candidat\_Offre.idOffre reference Offre.id  
 Domaine(id, nom)  
 Offre\_Domaine(idOffre, idDomaine, diplomeRecherche) Offre\_Domaine.idOffre reference Offre.id  
 Offre\_Domaine.idDomaine reference Domaine.id  
 Candidat\_Domaine(idCandidat, idDomaine, diplomePossede) Candidat\_Domaine.idCandidat reference Candidat.idPersonne  
 Candidat\_Domaine.idDomaine reference Domaine.id

### 3.4 Contraintes et Intégrités Référentielles

#### 3.4.1 Stratégies ON DELETE et ON UPDATE

##### 3.4.1.1 Candidat

**CONSTRAINT** FK\_Personne  
**FOREIGN KEY** (idPersonne)  
**REFERENCES** Personne(id)  
**ON DELETE CASCADE**  
**ON UPDATE CASCADE**

**CONSTRAINT** FK\_Adresse

```
FOREIGN KEY (idAdresse)
REFERENCES Adresse(id)
ON DELETE RESTRICT
ON UPDATE NO ACTION
```

- **Personne:** Si une personne est supprimée, le profil de candidat est également supprimé. Cela garantit qu'un candidat ne peut pas exister sans une identité personnelle de base.
- **Adresse:** L'adresse ne peut pas être supprimée si un candidat y est associé, afin de préserver l'intégrité des données historiques.

#### 3.4.1.2 Recruteur

```
FOREIGN KEY (idPersonne)
REFERENCES Personne(id)
ON DELETE CASCADE
ON UPDATE CASCADE
```

- Similaire au Candidat: si la personne est supprimée, son profil de recruteur est également supprimé.

#### 3.4.1.3 Recruteur\_Candidat

```
CONSTRAINT FK_Recruteur
FOREIGN KEY (idRecruteur)
REFERENCES Recruteur(idPersonne)
ON DELETE RESTRICT
ON UPDATE CASCADE
```

```
CONSTRAINT FK_Candidat
FOREIGN KEY (idCandidat)
REFERENCES Candidat(idPersonne)
ON DELETE RESTRICT
ON UPDATE CASCADE
```

- Empêche la suppression d'un recruteur ou d'un candidat s'il existe des relations professionnelles entre eux.
- Autorise la mise à jour des identifiants si nécessaire.

#### 3.4.1.4 Interactions (Emails, Appels, Entretiens)

```
FOREIGN KEY (idInteraction)
REFERENCES Interaction(id)
ON DELETE CASCADE
ON UPDATE CASCADE
```

- Les interactions spécialisées (email, appel, entretien) sont supprimées si l'interaction parent est supprimée.
- Permet de maintenir la cohérence des types d'interactions.

#### 3.4.1.5 Recruteur\_Interaction et Candidat\_Interaction

```
CONSTRAINT FK_Recruteur/Candidat
FOREIGN KEY (idRecruteur/idCandidat)
REFERENCES Recruteur/Candidat(idPersonne)
ON DELETE RESTRICT
ON UPDATE CASCADE
```

```
CONSTRAINT FK_Interaction
FOREIGN KEY (idInteraction)
REFERENCES Interaction(id)
ON DELETE RESTRICT
ON UPDATE CASCADE
```

- Empêche la suppression d'un recruteur ou candidat s'il a des interactions historiques.
- Interdit la suppression d'interactions liées à des personnes.

#### 3.4.1.6 Offre

```
CONSTRAINT FK_Adresse
FOREIGN KEY (idAdresse)
REFERENCES Adresse(id)
ON DELETE RESTRICT
ON UPDATE NO ACTION
```

- L'adresse associée à une offre ne peut pas être supprimée, préservant la localisation historique de l'offre.

#### 3.4.1.7 Contrat\_Travail

```
CONSTRAINT FK_Contrat_Travail
FOREIGN KEY (idOffre)
REFERENCES Offre(id)
ON DELETE RESTRICT
ON UPDATE NO ACTION
```

- Un contrat ne peut pas être lié à une offre supprimée, garantissant l'intégrité des références.

#### 3.4.1.8 Candidat\_Offre

```
CONSTRAINT FK_Candidat
FOREIGN KEY (idCandidat)
REFERENCES Candidat(idPersonne)
ON DELETE RESTRICT
ON UPDATE CASCADE
```

```
CONSTRAINT FK_Offre
FOREIGN KEY (idOffre)
REFERENCES Offre(id)
ON DELETE RESTRICT
ON UPDATE NO ACTION
```

- Empêche la suppression d'un candidat ayant postulé à des offres.
- Interdit la modification de la clé de l'offre une fois les candidatures enregistrées.

#### 3.4.1.9 Offre\_Domaine et Candidat\_Domaine

```
CONSTRAINT FK_Offre/Candidat
FOREIGN KEY (idOffre/idCandidat)
REFERENCES Offre/Candidat(id)
ON DELETE RESTRICT
ON UPDATE NO ACTION
```

```
CONSTRAINT FK_Domaine
FOREIGN KEY (idDomaine)
REFERENCES Domaine(id)
ON DELETE RESTRICT
ON UPDATE NO ACTION
```

- Préserve l'intégrité des liens entre offres/candidats et domaines.
- Empêche la suppression de domaines sans considérer leurs relations existantes.

Ces choix de contraintes visent à:

- Maintenir l'intégrité référentielle
- Prévenir les suppressions accidentelles
- Conserver un historique fiable
- Permettre des mises à jour contrôlées des données

### 3.4.2 Triggers de Validation

#### 3.4.2.1 check\_personne()

- **Objectif:** Garantir qu'une personne ne peut être créée que si un Candidat ou un Recruteur est aussi créé dans la transaction
- **Déclenchement:** Après l'insertion dans la table Personne
- **Action:** Lève une exception si la condition n'est pas respectée

#### 3.4.2.2 check\_personne\_exists()

- **Objectif:** Garantir qu'un Candidat ou un Recruteur ne peut être créé que si l'identifiant de personne existe déjà
- **Déclenchement:** Après l'insertion dans les tables Candidat et Recruteur
- **Vérification:** Vérifie que l'idPersonne n'est pas null et existe dans la table Personne
- **Action:** Lève une exception si la condition n'est pas respectée

#### 3.4.2.3 check\_interaction()

- **Objectif:** Garantir qu'une interaction ne peut être créée que si une interactions spécialisées (Email, Appel, Entretien) est aussi créée dans la transaction
- **Déclenchement:** Après l'insertion dans la table Interaction
- **Action:** Lève une exception si la condition n'est pas respectée



#### 3.4.2.4 `check_interaction_exists()`

- **Objectif:** Assurer que les interactions spécialisées (Email, Appel, Entretien) sont liées à une interaction principale existante
- **Déclenchement:** Après l'insertion dans les tables `Interaction_Email`, `Interaction_Appel`, `Interaction_Entretien`
- **Vérification:** Confirme que l'`idInteraction` n'est pas null et existe dans la table `Interaction`
- **Action:** Lève une exception si la condition n'est pas respectée

#### 3.4.2.5 `check_contrat_embauche()`

- **Objectif:** Empêcher la création d'un contrat de travail sans candidat embauché
- **Déclenchement:** Avant l'insertion dans la table `Contrat_Travail`
- **Vérification:** Vérifie qu'il existe au moins un candidat avec le statut 'Embauché' pour l'offre
- **Action:** Lève une exception si aucun candidat n'a été embauché

#### 3.4.2.6 `check_fin_after_cloture()`

- **Objectif:** Garantir que la date de fin du contrat est postérieure à la date de clôture de l'offre
- **Déclenchement:** Avant l'insertion ou la mise à jour dans la table `Contrat_Travail`
- **Vérification:** Compare la date de fin du contrat avec la date de clôture de l'offre
- **Action:** Lève une exception si la date de fin est antérieure ou égale à la date de clôture

#### 3.4.2.7 `check_candidat_offre_constraints()`

- **Objectif:** Valider les candidatures selon plusieurs contraintes
  1. Un statut 'Embauché' nécessite une date de clôture d'offre
  2. La date de postulation doit être entre la date de publication et la date de clôture de l'offre
- **Déclenchement:** Avant l'insertion ou la mise à jour dans la table `Candidat_Offre`
- **Vérification:**
  - Vérifie la cohérence du statut 'Embauché' avec la date de clôture
  - Contrôle que la date de postulation est dans la période valide de l'offre
- **Action:** Lève une exception si l'une des conditions n'est pas respectée

#### 3.4.2.8 `check_domaine_link()`

- **Objectif:** S'assurer qu'un domaine est lié soit à un candidat, soit à une offre
- **Déclenchement:** Après l'insertion ou la mise à jour dans la table `Domaine`
- **Vérification:** Compte les liens avec des candidats et des offres
- **Action:** Lève une exception si aucun lien n'est trouvé

Tous ces triggers visent à maintenir la cohérence et l'intégrité des données du système de gestion de candidatures.

### 3.4.3 Contraintes Métier

#### 3.4.3.1 Validation d'Âge des Candidats

**CHECK** (age >= 16 AND age < 100)

- **Objectif:** Garantir un intervalle d'âge réaliste pour les candidats
- **Plage:** Entre 16 et 99 ans
- **Logique:** Exclut les candidats potentiellement trop jeunes ou irréaliment âgés

#### 3.4.3.2 Validation des Coordonnées Géographiques (Adresse)

**CHECK** (latitude > -90 AND latitude < 90)

**CHECK** (longitude > -180 AND longitude < 180)

- **Objectif:** Assurer des coordonnées géographiques valides
- **Latitude:** Comprise entre -90° et 90° (pôles Nord et Sud)
- **Longitude:** Comprise entre -180° et 180° (méridien de Greenwich)
- **Logique:** Correspond aux limites physiques de la Terre

#### 3.4.3.3 Validation des Dates de Contrat de Travail

**CHECK** (fin IS NULL OR fin > debut)

- **Objectif:** Garantir la cohérence des dates de contrat
- **Condition:** Si une date de fin est spécifiée, elle doit être postérieure à la date de début
- **Flexibilité:** Permet des contrats sans date de fin (contrats indéterminés)

#### 3.4.3.4 Validation du Salaire Horaire

**CHECK** (salaireHoraire > 0)

- **Objectif:** Assurer un salaire horaire positif
- **Condition:** Le salaire doit être strictement supérieur à zéro
- **Logique:** Exclut les valeurs nulles ou négatives

#### 3.4.3.5 Validation des Dates de Publication et Clôture d'Offre

**CHECK** (dateCloture IS NULL OR dateCloture > datePublication)

- **Objectif:** Garantir la logique temporelle des offres
- **Condition:** Si une date de clôture existe, elle doit être postérieure à la date de publication
- **Flexibilité:** Autorise des offres sans date de clôture

## 4 Application

Dans cette section, nous présentons la partie applicative du projet.

L'application est contenue dans un backend, qui sert des pages statiques, et qui communique avec la base de donnée. Nous avons également un script qui génère un fichier "seed" SQL avec des données aléatoires.

### 4.1 Technologies Utilisées

Pour la réalisation de notre application nous avons utilisé la librairie Python **FastAPI** avec des templates HTML et CSS.

Nous avons utilisé la base de donnée PostgreSQL et containerisé avec Docker.

### 4.2 Structure de l'Application

Le projet contient les répertoires suivants:

- **docs/**: contient le schéma EA et le rapport.
- **app/**: contient le cœur de notre application.
- **generator/**: contient un script permettant de générer les données aléatoires.

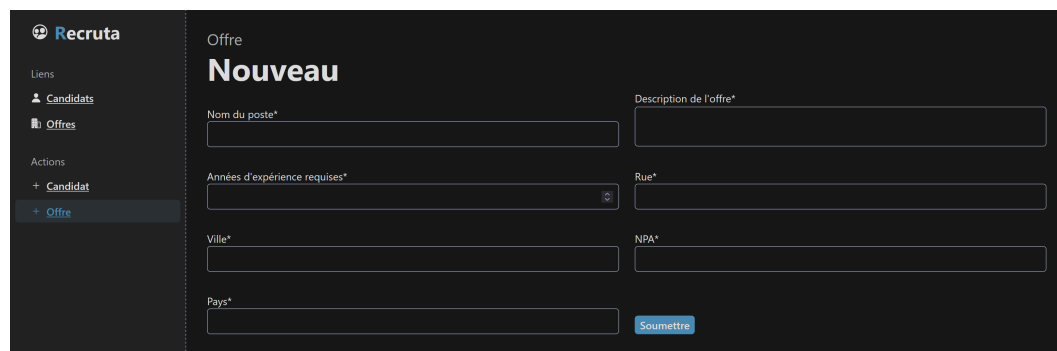
### 4.3 Manuel Utilisateur

#### 4.3.1 Introduction

Les sections suivantes visent à expliquer comment créer, modifier, consulter et filtrer les données que notre application permet de visualiser. Au premier démarrage de l'application une quantité suffisante de données sera générée et automatiquement ajoutée à la base de données. Ceci permettant de tester les différentes fonctionnalités sans avoir à créer chaque offre ou candidat à la main.

#### 4.3.2 Création d'une Offre

Pour ajouter une offre dans la base de données, il faut se rendre sur <http://localhost:8000/offre-new> ou cliquer sur **+ Offre**. Afin d'être redirigé sur la page suivante.



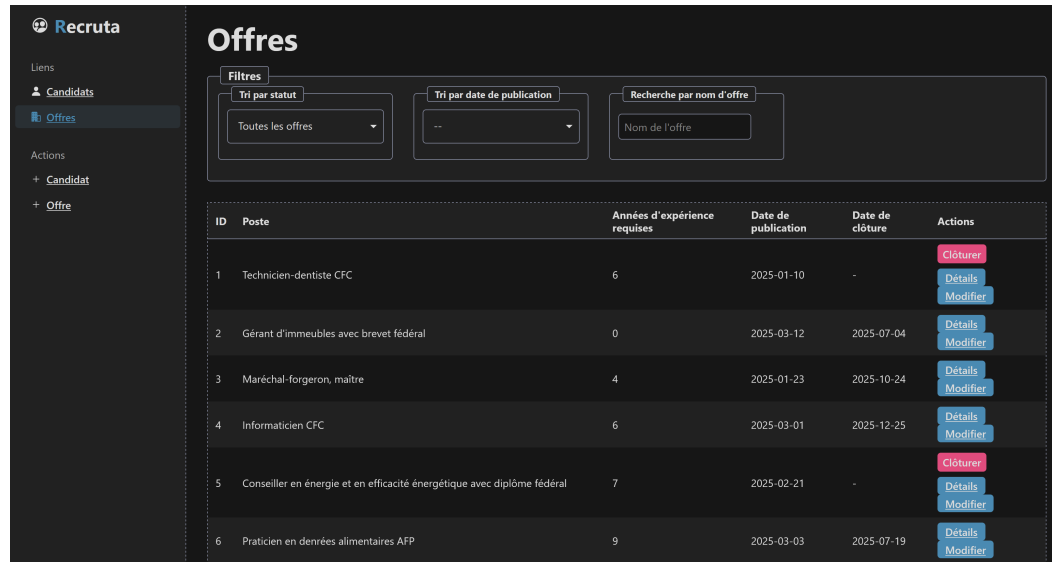
The screenshot shows the 'Nouvelle offre' (New Offer) page in the Recruta application. The sidebar on the left contains the following links: 'Liens', 'Candidats', 'Offres', 'Actions', '+ Candidat', and '+ Offre' (highlighted). The main form area is titled 'Offre Nouveau' and contains the following fields: 'Nom du poste\*', 'Description de l'offre\*', 'Années d'expérience requises\*' (with a dropdown arrow), 'Rue\*', 'Ville\*', 'NPA\*', and 'Pays\*'. A 'Soumettre' button is located at the bottom right of the form.

Figure 2: Nouvelle offre

On peut renseigner les différents champs nécessaires à la création de l'offre puis cliquer sur soumettre.

#### 4.3.3 Liste des Offres

La liste de toutes les offres se trouve à l'url <http://localhost:8000/offres> que l'on peut accéder en cliquant sur **Offres**.



The screenshot shows the 'Offres' page in the Recruta application. The page has a dark theme. On the left is a sidebar with 'Recruta' logo, 'Liens', 'Candidats', 'Offres' (selected), and 'Actions' with '+ Candidat' and '+ Offre'. The main area is titled 'Offres' and contains a 'Filtres' section with three filters: 'Tri par statut' (set to 'Toutes les offres'), 'Tri par date de publication' (set to '--'), and 'Recherche par nom d'offre' (with a search input). Below the filters is a table of job offers.

ID	Poste	Années d'expérience requises	Date de publication	Date de clôture	Actions
1	Technicien-dentiste CFC	6	2025-01-10	-	<a href="#">Clôturer</a> <a href="#">Détails</a> <a href="#">Modifier</a>
2	Gérant d'immeubles avec brevet fédéral	0	2025-03-12	2025-07-04	<a href="#">Détails</a> <a href="#">Modifier</a>
3	Maréchal-forgeron, maître	4	2025-01-23	2025-10-24	<a href="#">Détails</a> <a href="#">Modifier</a>
4	Informaticien CFC	6	2025-03-01	2025-12-25	<a href="#">Détails</a> <a href="#">Modifier</a>
5	Conseiller en énergie et en efficacité énergétique avec diplôme fédéral	7	2025-02-21	-	<a href="#">Clôturer</a> <a href="#">Détails</a> <a href="#">Modifier</a>
6	Praticien en denrées alimentaires AFP	9	2025-03-03	2025-07-19	<a href="#">Détails</a> <a href="#">Modifier</a>

Figure 3: Offres

On peut observer certaines informations sur les offres. Il est également possible de clôturer une offre si elle est encore ouverte, en cliquant sur **Clôturer**, ce qui lui assignera la date d'aujourd'hui comme date de clôture.

#### 4.3.4 Détail d'une Offre

Depuis la page précédente en cliquant sur **Détails** on accède à <http://localhost:8000/offres/id> avec id le numéro correspondant à l'id de l'offre.

On obtient ainsi des informations supplémentaires sur l'offre ainsi que la liste des candidats qui y ont postulé. On a accès au statut des candidats et à leur score de compatibilité avec l'offre. Le bouton **Tri par score** permet de trier les candidats ayant postulé à cette offre du meilleur au moins bon score de compatibilité.

#### 4.3.5 Modification d'une Offre

Pour modifier les informations d'une offre il faut se rendre sur <http://localhost:8000/offre/id/edit> avec id le numéro correspondant à l'id de l'offre que l'on souhaite modifier. Cette page est accessible depuis les différents boutons **Modifier** que l'on voit dès qu'une offre est affichée à différents endroits de l'application.

Recruta

Liens

Candidats

Offres

Actions

+ Candidat

+ Offre

Offre

Technicien-dentiste CFC

La liberté de louer autrement

Couverts

ID	Années d'expérience requises	Date de publication	Date de clôture	Rue	Ville	NPA	Pays
1	6	2025-01-10	-	boulevard Duroux 594	Beuret	7854	Suisse

Domaine

Diplôme Recherché

Enseignement

Master

Modifier

Voir les candidats

Tri par score

Candidats

ID	Statut	Date de postulation	Nom	Prénom	Email	Âge	Genre	Numéro de téléphone	Années d'expérience	Score	Actions
45	En cours	2025-02-27	Crivelli	Patrick	besanconolivier@example.com	18	Femme	+41 82 503 16 94	36	0	<div>Détails</div> <div>Modifier</div>
50	En cours	2025-01-17	Chenaux	Mélanie	aurelie92@example.org	41	Autre	+41 45 902 27 45	27	0	<div>Détails</div> <div>Modifier</div>
73	En cours	2025-02-09	Mottiez	Anna	gilbert42@example.com	30	Femme	0901 974 098	14	0	<div>Détails</div> <div>Modifier</div>
81	Refusé	2025-01-30	Mottet	Ethan	mmottet@example.net	58	Femme	+41 (0)62 879 68 59	20	0	<div>Détails</div> <div>Modifier</div>
96	Refusé	2025-01-12	Beuchat	Marguerite	bernardberberat@example.org	40	Autre	024 005 91 16	9	0	<div>Détails</div> <div>Modifier</div>

Figure 4: Détail offre

Recruta

Liens

Candidats

Offres

Actions

+ Candidat

+ Offre

Offre

Modifier

Nom du poste\*

Technicien-dentiste CFC

Description de l'offre\*

La liberté de louer autrement

Années d'expérience requises\*

6

Rue\*

boulevard Duroux 594

Ville\*

Beuret

NPA\*

7854

Pays\*

Suisse

Soumettre

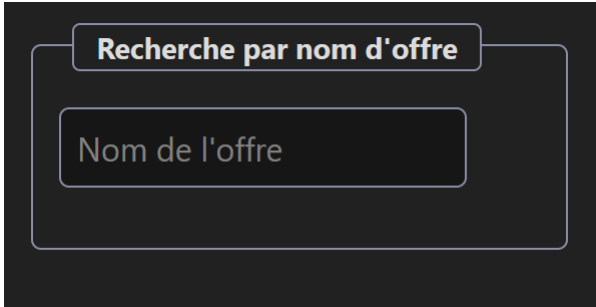
Figure 5: Modifier Offre

Cela nous redirige sur une page similaire à celle de création d'offre mais avec les champs pré remplis contenant les informations actuelles de l'offre à modifier. Une fois les champs modifier, il suffit de cliquer sur **Soumettre**.

#### 4.3.6 Filtre des Offres

La page <http://localhost:8000/offres> permet de filtrer les offres en y ajoutant différent query parameters. Soit directement dans l'URL, soit à l'aide des différents selecteurs fournis.

Interface	Query parameters
	<code>?statut=all,</code> <code>?statut=open,</code> <code>?statut=closed</code>
	<code>?datePublication=recent</code> <code>?datePublication=oldest</code>

Interface	Query parameters
	<code>?name={name}</code> <code>?idCandidat={id}</code>
<p>Depuis la page <a href="http://localhost:8000/candidats/id">http://localhost:8000/candidats/id</a>, cliquer sur <b>Voir les postulations</b> permet de ne voir que les offres auxquelles un certain candidat a postulé.</p>	

Il est possible de combiner les filtres en utilisant les différents outils de l'interface ou en séparant les query parameters par des `&`. Notons cependant qu'il ne faut pas avoir plusieurs fois le même query parameter avec des valeurs différentes. La barre de recherche permet de chercher une offre par nom, pour les offres dont le nom est composé de plusieurs mots, cela permet de les retrouver avec n'importe lequel des mots qui la compose. Par exemple l'offre Technicien-dentiste pourra être retrouvée en recherchant Technicien, dentiste, ou encore Technicien-dentiste.

#### 4.3.7 Création d'un Candidat

Pour ajouter un candidat dans la base de données, il faut se rendre sur <http://localhost/candidat-new> que l'on peut atteindre en cliquant sur **+ Candidat**

On remplit les champs requis, et l'on clique sur **Soumettre**. Notons que pour assigner ce candidat à une offre il faut que l'offre ait été créée précédemment afin qu'elle figure dans la liste des offres proposées. Il est possible de l'assigner à plusieurs offres en maintenant la touche **CTRL** en cliquant sur les différentes offres souhaitées.

#### 4.3.8 Liste des Candidats

La liste de tous les candidats se trouve sur <http://localhost:8000/candidats> que l'on peut accéder en cliquant sur **Candidats**.

On peut observer différentes informations sur tous les candidats de la base de données.

#### 4.3.9 Détail d'un Candidat

Depuis la page précédente en cliquant sur **Détails** on accède à <http://localhost:8000/candidats/id> avec `id` le numéro correspondant à l'id du candidat.

Cette page nous donne des informations supplémentaires sur le candidat ainsi que la liste des offres auquel il a postulé et son statut les concernant.

Recruta

Liens

- Candidats
- Offres

Actions

- + Candidat
- + Offre

Candidat

Nouveau

Nom\*

Prénom\*

Email\*

Âge\*

Genre\*

Homme

Numéro de téléphone\*

Années d'expérience\*

Rue\*

Ville\*

NPA\*

Pays\*

Offres\* (CTRL pour en selectionner plusieurs)

Technicien-dentiste CFC - La liberté de louer autrement  
Gérant d'immeubles avec brevet fédéral - Le plaisir de louer sans soucis  
Maréchal-forgeron, maître - L'assurance d'évoluer de manière efficace  
Informaticien CFC - Le plaisir d'avancer de manière efficace

Soumettre

Figure 6: Nouveau candidat

Recruta

Liens

- Candidats
- Offres

Actions

- + Candidat
- + Offre

Candidats

Filtres

Genres

- Homme
- Femme
- Autre

Age

Min Age: 16

Max Age: 100

Expérience

Min Exp.

Max Exp.

ID	Nom	Prénom	Email	Âge	Genre	Numéro de téléphone	Années d'expérience	Actions
11	Vuille	Catherine	besanconnolan@example.com	63	Autre	+41 49 876 98 12	1	Détails Modifier
12	Vallélian	Ethan	emma43@example.net	50	Autre	+41 21 033 35 81	40	Détails Modifier
13	Rossellat	Nathalie	aevequoz@example.net	60	Homme	+41 42 672 87 12	35	Détails Modifier
14	Rosselet	Noah	carronsuzanne@example.net	54	Autre	0846 504 415	28	Détails Modifier
15	Mayor	Marcel	rosseletchantal@example.net	64	Femme	+41 (0)65 837 42 73	1	Détails Modifier
16	Cattin	Sara	rmonnard@example.org	60	Femme	+41 90 448 01 75	24	Détails Modifier
17	Candaux	Ana	georges29@example.net	52	Femme	057 070 22 48	24	Détails Modifier
18	Rossellat	Roger	cosandeychloe@example.com	45	Autre	+41 (0)52 866 06 33	14	Détails Modifier
19	Chaudet	François	nsansonnens@example.net	39	Homme	+41 91 470 41 27	19	Détails Modifier
20	Blanc	Gabriel	pascal00@example.com	36	Homme	+41 (0)97 136 91 33	7	Détails Modifier
21	Bouvier	Paul	elodie09@example.com	49	Femme	+41 (0)67 287 07 63	21	Détails Modifier
22	Mayor	Jonathan	qniquille@example.org	62	Autre	+41 (0)24 532 54 78	0	Détails Modifier
23	Morard	Stéphane	mcarraux@example.org	29	Femme	+41 54 446 42 92	19	Détails Modifier

Figure 7: Candidats



The screenshot shows the 'Recruta' application interface. On the left is a sidebar with navigation links: 'Liens', 'Candidats', 'Offres', and 'Actions'. The main content area is titled 'Candidat Vallélian Ethan'. Below the title is a table with personal information: ID (12), Nom (Vallélian), Prénom (Ethan), Email (emma43@example.net), Âge (50), Genre (Autre), Numéro de téléphone (+41 21 033 35 81), Années d'expérience (40), Rue (chemin Christine Bernasconi 65), Ville (Comman-sur-Tinguely), NPA (1464), and Pays (Suisse). Below this is a table for 'Domaine' and 'Diplôme'. The 'Domaine' table has two rows: 'Developpement informatique' and 'Marketing'. The 'Diplôme' table has two rows: 'Master' and 'Aucun'. There are two buttons: 'Modifier' and 'Voir les postulations'. Below this is a section titled 'Offres' with a table of job offers. The table has columns: ID, Date de postulation, Statut, Poste, Date de publication, Date de clôture, Score, and Actions. There are two rows of offers. The first row has ID 22, Date 2025-05-26, Statut 'Refusé', Poste 'Agent professionnel de protection de personnes et de biens avec brevet fédéral', Date de publication 2025-05-11, Date de clôture 2025-06-17, Score 0, and Actions 'Détails' and 'Modifier'. The second row has ID 42, Date 2025-06-17, Statut 'Refusé', Poste 'dipl. in Unternehmens- und Mitarbeiterführung NDS HF', Date de publication 2025-04-21, Date de clôture '-', Score 50, and Actions 'Clôturer', 'Détails', and 'Modifier'.

ID	Nom	Prénom	Email	Âge	Genre	Numéro de téléphone	Années d'expérience	Rue	Ville	NPA	Pays
12	Vallélian	Ethan	emma43@example.net	50	Autre	+41 21 033 35 81	40	chemin Christine Bernasconi 65	Comman-sur-Tinguely	1464	Suisse

Domaine	Diplôme
Developpement informatique	Master
Marketing	Aucun

ID	Date de postulation	Statut	Poste	Date de publication	Date de clôture	Score	Actions
22	2025-05-26	Refusé	Agent professionnel de protection de personnes et de biens avec brevet fédéral	2025-05-11	2025-06-17	0	Détails Modifier
42	2025-06-17	Refusé	dipl. in Unternehmens- und Mitarbeiterführung NDS HF	2025-04-21	-	50	Clôturer Détails Modifier

Figure 8: Détail candidat

#### 4.3.10 Modification d'un Candidat

Pour modifier un candidat il faut se rendre sur `http://localhost:8000/candidat/id/edit` avec id le numéro correspondant à l'id du candidat que l'on souhaite modifier. On peut facilement atteindre cette page depuis les divers boutons **Modifier** présents à chaque fois que l'on affiche des informations pour un candidat.

Cela nous amène sur un formulaire similaire à celui de la création d'un nouveau candidat mais pour lequel les champs sont pré remplis avec les informations actuelles du candidat. Une fois les changement souhaités effectués, on clique sur **Soumettre** pour les rendre effectifs.

#### 4.3.11 Filtre des Candidats

La page `http://localhost:8000/candidats` permet de filtrer les candidats en y ajoutant différent query parameters. Soit directement dans l'URL, soit à l'aide des différents sélecteurs fournis.

**Recruta**

Liens

- Candidats
- Offres

Actions

- + Candidat
- + Offre

**Candidat**

## Modifier

Nom\* Vallélian

Prénom\* Ethan

Email\* emma43@example.net

Âge\* 50

Genre\* Autre

Numéro de téléphone\* +41 21 033 35 81

Années d'expérience\* 40

Rue\* chemin Christine Bernasconi 65

Ville\* Comman-sur-Tinguely

NPA\* 1464

Pays\* Suisse

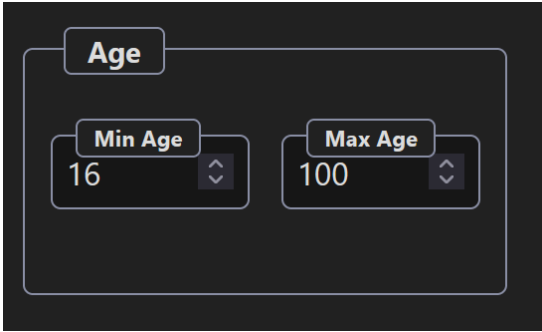
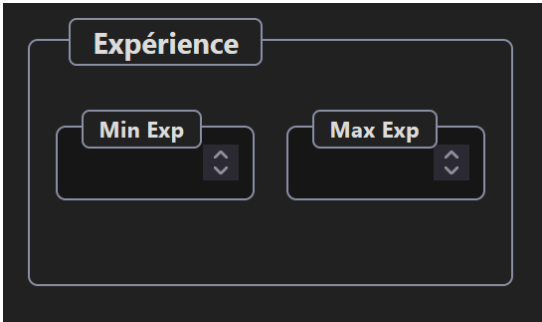
Offres\* (CTRL pour en sélectionner plusieurs)

- Gérant d'immeubles avec brevet fédéral - Le plaisir de louer sans soucis
- Maréchal-forgeron, maître - L'assurance d'évoluer de manière efficace
- Informaticien CFC - Le plaisir d'avancer de manière efficace
- Conseiller en énergie et en efficacité énergétique avec diplôme fédéral - Le confort d'innover de manière sûre
- Praticien en denrées alimentaires AFP - Le plaisir d'avancer en toute tranquillité

Soumettre

Figure 9: Modifier candidat

Interface	Query parameters
<p><b>Genres</b></p> <p><input type="radio"/> Homme</p> <p><input type="radio"/> Femme</p> <p><input type="radio"/> Autre</p>	<p>?gender=Homme, ?gender=Femme, ?gender=Autre pouvant être combinés pour avoir plusieurs genres.</p>

Interface	Query parameters
	<code>?minAge={minAge}&amp;maxAge={maxAge}</code> Si une valeur n'est pas fournie, elle est remplacée par une valeur par défaut : minAge=16 et maxAge=100
	<code>?minExp={minExp}, ?maxExp={maxExp} ?idoffre={id}</code>

Depuis la page `http://localhost:8000/offres/id`, cliquer sur **Voir les candidats** permet de ne voir que les candidats ayant postulé à une certaine offre.

Il est possible de combiner les filtres en utilisant les différents outils de l'interface ou en séparant les query parameters par des `&`. Notons cependant qu'il ne faut pas avoir plusieurs fois le même query parameter avec des valeurs différentes, à l'exception de `?gender`

## 5 Conclusion

Pour conclure, dans le cadre de ce projet, nous avons conceptualisé et réalisé une application permettant de faciliter la visualisation et la recherche de candidats pour une offre d'emploi.

Malgré le temps limité dont nous disposions, nous avons réussi à implémenter de nombreuses fonctionnalités en mettant l'accent sur la simplicité et l'efficacité.

Cependant, certaines fonctionnalités prévues n'ont pas pu être développées. Pour aller plus loin, nous pourrions ajouter des interactions directes entre candidats et recruteurs, ainsi qu'approfondir les mécanismes de matching entre candidats et offres.

## A Annexes

### A.1 Guide d'Installation et d'Utilisation

Ci-dessous sont les étapes à suivre afin de démarrer et utiliser l'application.

#### A.1.1 Avec Docker

Afin de démarrer l'application, naviguez jusqu'au sous-dossier `app` et lancez les conteneurs :

```
docker-compose up --build
```

Ceci va lancer la base de données, la populer et démarrer le serveur web sur le port : `http://localhost:8000`.

#### A.1.2 Sans Docker

Afin d'installer les dépendances en local, commencez par créer un environnement virtuel :

```
python3 -m venv .venv  
source .venv/bin/activate
```

Puis installez les dépendances :

```
pip install -r requirements.txt
```