



---

SEARCH JOBS

---

---

# SOMMAIRE

---

Search jobs .....	1
Cahier des charges .....	3
Modele conceptuel de données.....	4
Modèle relationnel de données :.....	5
Schémas visuel .....	5
Architecture de l'application.....	6
Principe de développement et outils .....	7
Découpage de l'application.....	9
Section 1 : L'accueil .....	9
Section 2 : Listing des offres.....	9
Section 3 Détail d'une offre.....	9
Section 4 Inscription.....	9
Section 5 Mon compte .....	9
Section 6 Ajout d'une offre .....	10



---

# CAHIER DES CHARGES

---

Search jobs est une application web qui a pour but de permettre à ses utilisateurs de rechercher des emplois dans le secteur de l'informatique une fois ces derniers authentifier.

Pour cette première version, la page d'accueil contiendra un formulaire de recherche permettant de chercher une offre par mots clefs ou/et lieu. Une page offre doit permettre de lister toutes les offres du site par date de publications, de la plus récente à la plus ancienne. Les détails des offres n'est visible que pour les utilisateurs connectés. Les utilisateurs peuvent s'inscrire via un formulaire d'inscription placé sur une page d'inscription.

Une fois connecté, en fonction de son statut l'utilisateur aura plusieurs possibilités.

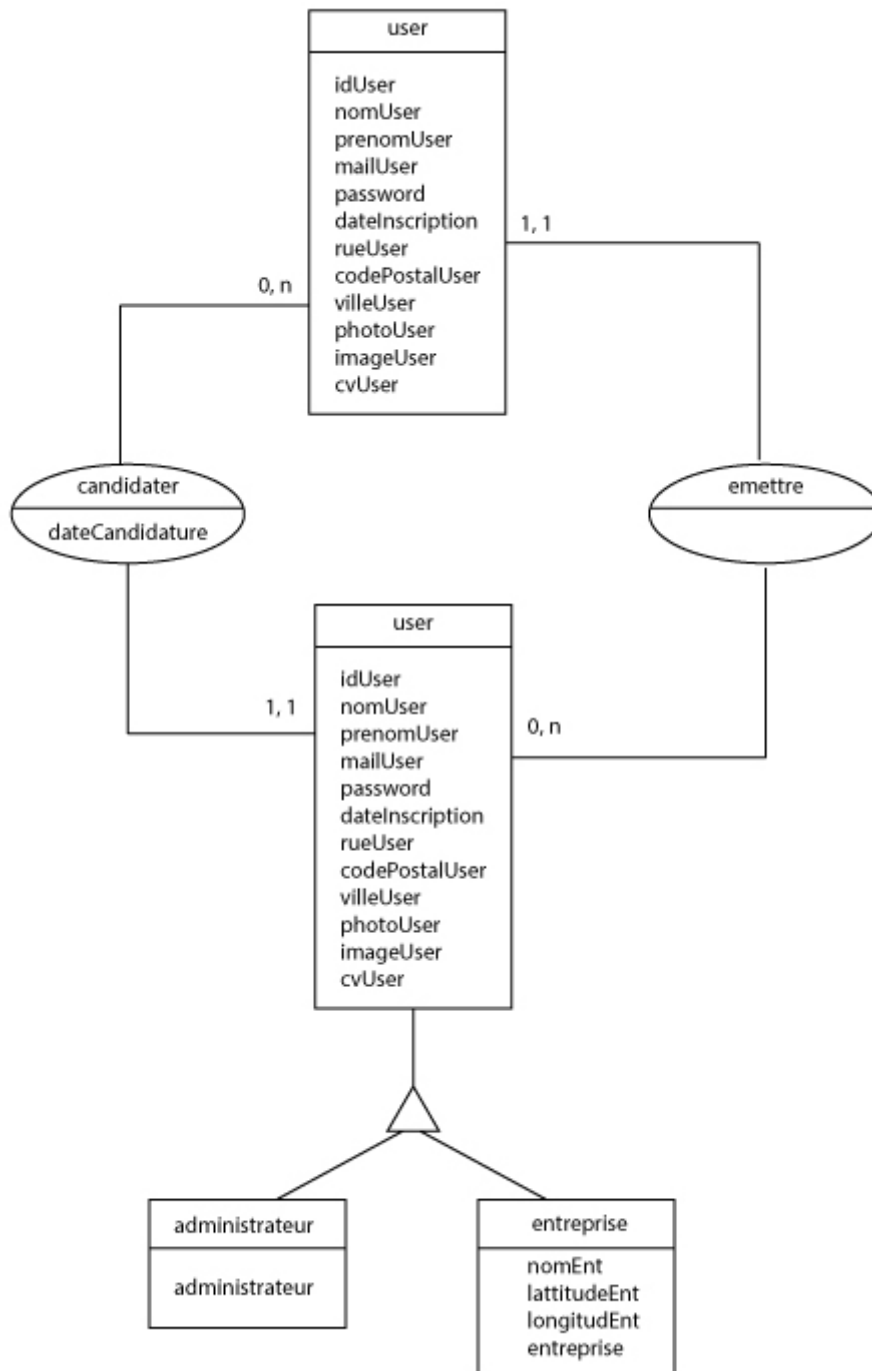
S'il a le statut de demandeur il pourra :

- Voir le détail des offres d'emplois avec la possibilité de sélectionner les offres qui l'intéressent.
- Voir le listing des offres qu'il a sélectionné.
- Voir son profil.

S'il a le statut d'entreprise, il pourra :

- Voir le détail des offres d'emplois
- Voir son profil
- Ajouter une offre
- Voir quels sont les demandeurs qui ont sélectionner les offres afin de visualiser leur profil et consulter leurs CV.
- Voir le listing des offres qu'il a de visible sur le site.

# MODELE CONCEPTUEL DE DONNEES



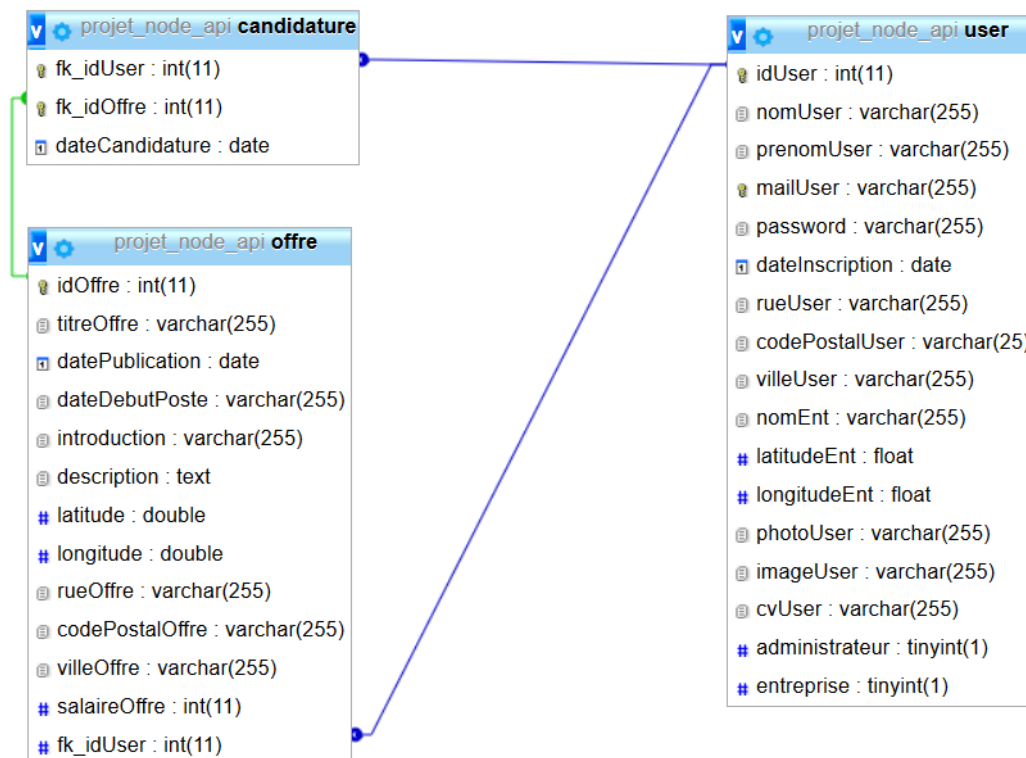
## Modèle relationnel de données :

offre (idOffre, titreOffre, datePublication, dateDebutPoste,description, latitude, longitude, salaire, #fk\_idUser)

candidature (#fk\_idOffre, #fk\_idUser, dateCandidature)

user (idUser, nomUser, prenomUser, mailUser, password, dateInscription, rueUser, codePostalUser, villeUser, nomEnt, latitudeEnt, longitudeEnt, photoUser, imageUser, cvUser, administrateur , entreprise)

## Schémas visuel



# ARCHITECTURE DE L'APPLICATION

L'application a été développée sous JavaScript avec le Framework Angular 5 pour la partie front, et Node.js pour la partie API.

Partie Angular :

```
src
├── app
│   ├── home
│   │   ├── home.component.html
│   │   ├── home.component.scss
│   │   └── home.component.ts
│   ├── not-found
│   │   ├── not-found.component.html
│   │   ├── not-found.component.scss
│   │   └── not-found.component.ts
│   ├── offres
│   │   ├── add-offre
│   │   ├── class
│   │   ├── details
│   │   ├── gestion-offres
│   │   ├── list-offres
│   │   ├── search-offres
│   │   ├── offres.module.ts
│   │   └── offres.service.ts
│   └── users
│       ├── authentication
│       ├── class
│       ├── compte
│       ├── compte-admin
│       ├── compte-firm
│       ├── compte-user
│       ├── gestion-user
│       ├── inscription
│       ├── statistiques
│       ├── user-details
│       ├── user-modification
│       ├── users.module.ts
│       └── users.service.ts
├── app-routing.module.ts
├── app.component.css
├── app.component.html
├── app.component.ts
└── app.module.ts
```

API :

```
api-jobs
├── configuration
│   ├── db.js
│   └── secret.js
├── models
│   ├── offres.js
│   └── users.js
├── node_modules
├── public
│   ├── uploads
│   │   ├── cv
│   │   ├── img
│   │   └── photo
│   ├── .gitignore
│   ├── LICENSE
│   ├── package-lock.json
│   ├── package.json
│   ├── README.md
│   └── server.js
```

---

# PRINCIPE DE DEVELOPPEMENT ET OUTILS

---

L'application a été développée sous la façon modulaire afin de rendre le plus possible réutilisable les modules et le composant qu'il comporte pour d'autres projets.

« L'équipe Angular a travaillé dur pour nous offrir un framework à l'architecture robuste et évolutive.

On retrouve ainsi les différents concepts que sont le data binding, les composants, les directives, les injectors, et les modules.

Les applications Angular se doivent d'être modulaires, et pour se faire, Angular utilise son propre système de modularité appelé NgModule.

Une application Angular possède au moins un module, le root module, nommé par convention AppModule. Mais bien souvent, elle possède plusieurs modules s'attachant, chacun, à proposer une fonctionnalité métier, une capacité technique, ou tout ce qui peut rendre votre code clair et modulaire

Un module Angular n'a rien à voir avec un module JavaScript ES6 du moins au niveau de la provenance : l'un vient de la norme EcmaScript, l'autre de l'équipe Angular. La norme ES6 (EcmaScript 6) JavaScript propose également son propre gestionnaire de module. Un module JavaScript ES6 est un fichier qui exporte quelque chose (une fonction, une classe ou un objet) que les autres fichiers pourront consommer.

De l'autre côté, un module Angular regroupe un ensemble de composants Angular dans une même unité logique. Cela peut sembler être un peu déroutant au début, mais il faut comprendre que ces modules sont liés à vos propres souhaits de découpage applicatif. Cela apporte une cohésion applicative. »

<http://www.learn-angular.fr/les-modules-angular/>

Concernant la partie API qui récupère les données, j'ai utilisé Node.js et le Framework Express.js.

Node.js® est une exécution JavaScript basée sur le moteur JavaScript V8 de Chrome . Node.js utilise un modèle d'E / S piloté par événement et non bloquant qui le rend léger et efficace. L'écosystème de paquets de Node.js, npm , est le plus grand écosystème de bibliothèques open source au monde

Express est une infrastructure d'applications Web Node.js minimaliste et flexible qui fournit un ensemble de fonctionnalités robuste pour les applications Web et mobiles. Grâce à une

foule de méthodes utilitaires HTTP et de middleware mise à notre disposition, la création d'une API robuste est simple et rapide.

L'ensemble des pages de cette application respecte les normes des w3c.

La base de données utilisée est MySQL.





---

# DECOUPAGE DE L'APPLICATION

---

## Section 1 : L'accueil

Au chargement de l'application, l'utilisateur est dirigé vers la page d'accueil. Sur celle-ci, il peut visualiser un message d'accueil et faire une recherche d'emploi. Il a également la possibilité d'accéder à la page d'inscription mais aussi à la page de connexion et de listing de toutes les offres.

## Section 2 : Listing des offres

La page de listing des offres comme son nom l'indique liste toutes les offres du site de la plus récente à la plus ancienne. L'utilisateur peut accéder aux détails de celle-ci en cliquant sur l'offre choisie.

Si l'utilisateur est connecté et qu'il a le statut de demandeur d'emplois, il peut sélectionner l'offre afin de notifier l'entreprise émettrice qu'il est intéressé par cette dernière.

## Section 3 Détail d'une offre

La page détail d'une offre affiche le contenu de l'offre et une google map pour situer cette dernière uniquement si l'utilisateur est connecté. Sinon seulement l'introduction est affichée ainsi qu'un message invitant l'utilisateur à ce connecter.

## Section 4 Inscription

La page d'inscription comme son nom l'indique, permet à une personne de s'enregistrer en tant que demandeur d'emploi ou entreprise.

## Section 5 Mon compte

Si l'utilisateur a le statut de demandeur d'emploi, la page Compte lui permet de voir la liste des offres qu'il a sélectionné et d'accéder à son profil.

Si l'utilisateur a le statut d'entreprise, il peut voir la liste de ses offres d'emploi présent sur le site mais aussi la liste des demandeurs d'emploi intéressé par ses offres. Il peut ainsi accéder à chaque profil intéressé et voir le CV des candidats. L'utilisateur a également avec ce statut la possibilité de rajouter une nouvelle offre d'emploi.

## Section 6 Ajout d'une offre

Cette page n'est visible que si l'utilisateur a le statut d'entreprise. Elle permet de rajouter une offre d'emploi sur le site.

