

COMPTE RENDU

---

# Projet Job'Stone

---

IDRISS BENGUEZZOU  
ANTOINE CHATAIGNIER

Mai 2021

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Présentation globale du projet</b>	<b>3</b>
2.1	Architecture . . . . .	3
2.2	Base de données . . . . .	4
2.2.1	USER . . . . .	4
2.2.2	POST . . . . .	4
2.2.3	CATEGORY et APPLY . . . . .	5
<b>3</b>	<b>Déroulé du projet</b>	<b>5</b>
3.1	Partage du travail . . . . .	5
3.2	Difficultés rencontrées . . . . .	6
3.2.1	Rôle de l'administrateur . . . . .	6
3.2.2	Publication du projet sur Webetu . . . . .	6
3.2.3	Problèmes en phase de développement . . . . .	6
3.2.3.a	Lien entre PHP et JS . . . . .	6
3.2.3.b	Redirection de pages . . . . .	6
<b>4</b>	<b>Job'Stone : une révolution</b>	<b>7</b>
4.1	L'architecture, le retour . . . . .	7
4.2	Direction : Les étoiles . . . . .	7
4.3	Vers la perfection grâce aux filtres . . . . .	7
4.4	Le destin filtrera . . . . .	7
4.5	PHP tu changeras, rapide tu seras . . . . .	8
4.6	Administration . . . . .	8
4.7	Pseudo ou mail, c'est du pareil au même . . . . .	8
<b>5</b>	<b>Conseils pour découvrir la plateforme</b>	<b>8</b>
<b>6</b>	<b>Conclusion</b>	<b>8</b>

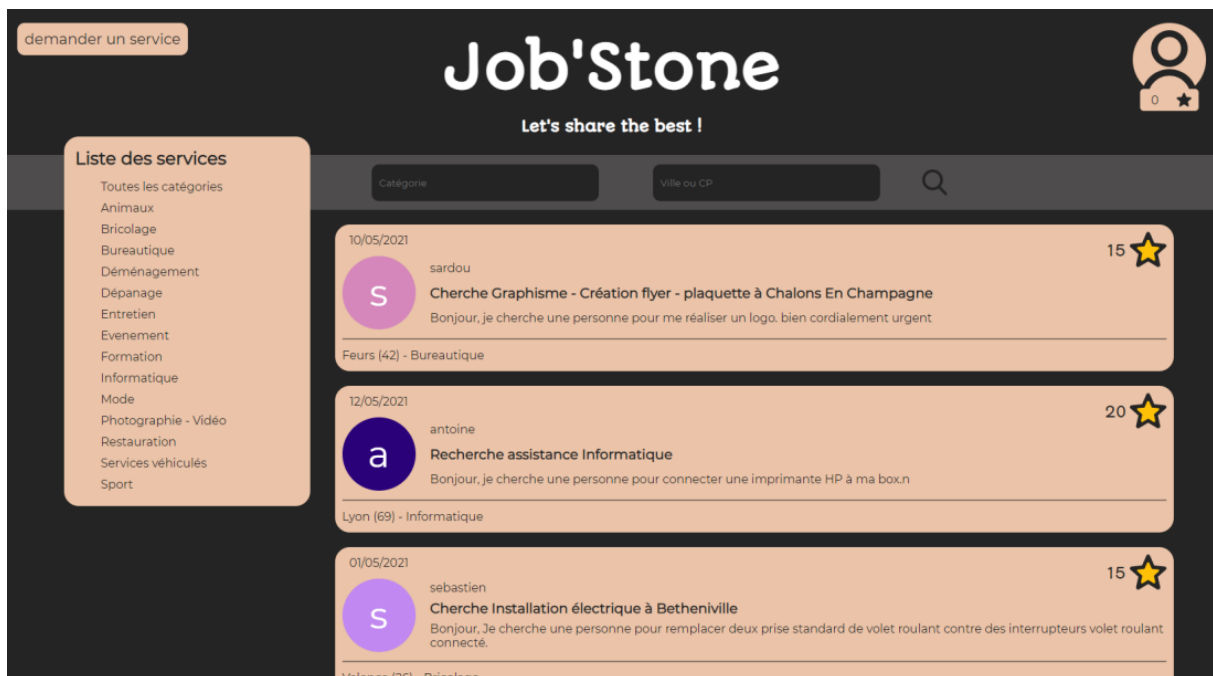
# 1 Introduction

Job'Stone est un site web d'entraide entre particuliers inspiré du célèbre site "Allo Voisin". Comme ce dernier, le site web nouvellement créé permet à des utilisateurs de poster des annonces ou de candidater à celles-ci dans un esprit d'entraide collectif, basé sur une idéologie simple : offrir de son temps.

Job'Stone permet à tout le monde de voir les annonces publiées, mais seul les utilisateurs connectés peuvent avoir accès aux fonctionnalités du site. Ainsi, après s'être inscrit, chacun sera libre de poster une annonce, ou de candidater à l'une d'entre elles. La différence marquante avec la source d'inspiration "Allo Voisin" reste le mode de paiement. Si Job'Stone peut envisager dans le futur, de devenir une plateforme incluant les paiements par carte bancaire, ce n'est pas le cas aujourd'hui. Chaque compte se voit attribuer à sa création un nombre d'étoiles qui permettra à l'utilisateur de payer pour des services. Pour gagner plus d'étoiles, l'utilisateur pourra s'il le souhaite, rendre des services à d'autres utilisateurs, permettant ainsi de rencontrer de nouvelles personnes en donnant un peu de son temps personnel.

Les annonces peuvent être filtrées, par catégorie ou par ville. L'objectif étant, bien entendu, de partager son temps avec la communauté proche, chaque service n'ayant pas vocation à durer sur le long terme ni à être un travail à plein temps.

La philosophie de Job'Stone ? Offrir sa force dans un domaine tout en recevant la force des autres. Le monde avance mieux à plusieurs.



## 2 Présentation globale du projet

Une fois l'idée de Job'Stone trouvée et mûrement réfléchie, il a fallut passer à la partie pratique : le développement du site web.

Nous avons pour ambition de séparer au maximum chaque langage du projet. C'est avec la partie HTML/CSS que nous avons commencé, travaillant sur une maquette du projet à laquelle nous pourrions greffer notre JS une fois le visuel terminé. C'est ce que nous avons fait ensuite, et la finalité du projet a été de rajouter les requêtes au serveur et à la base de données en PHP.

### 2.1 Architecture

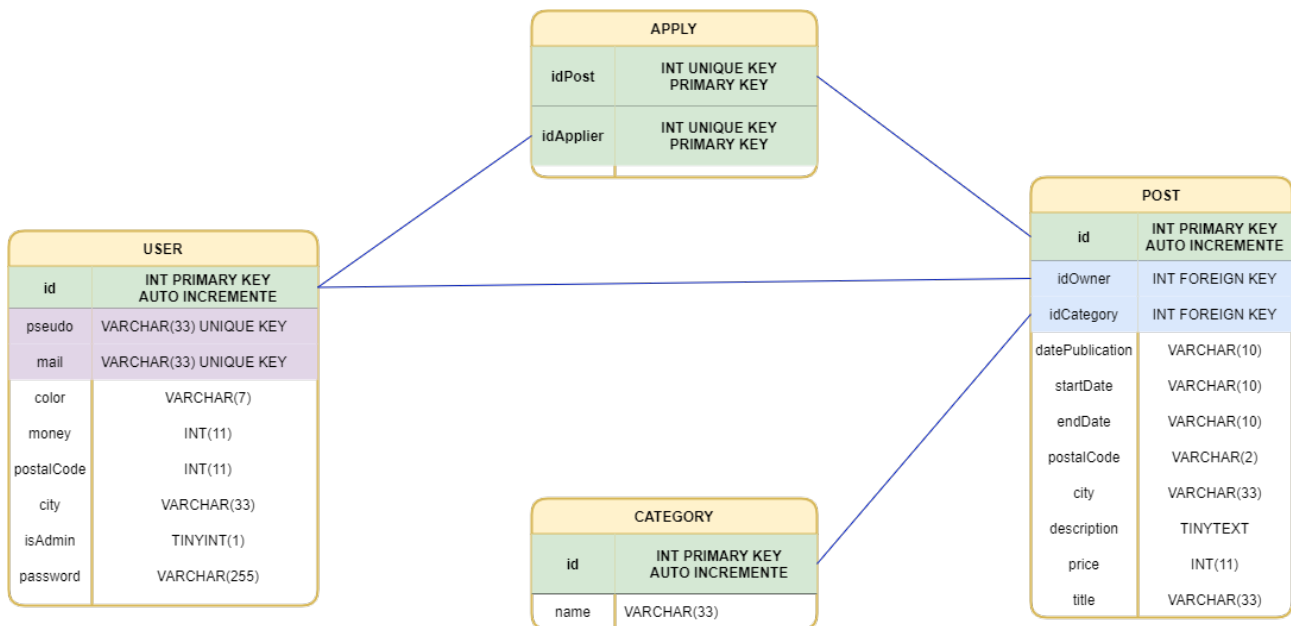
Le projet est donc aujourd'hui constitué de cinq dossiers principaux :

- **le dossier "pages"** : La partie concentrant nos pages HTML.
- **le dossier "styles"** : La partie contenant les styles. Dans ce dossiers se retrouve un dossier pour chaque page, comprenant le style "general" de la page, puis plus tard les styles "phone" et "computer", spécifiques à un type d'appareil : mobile ou ordinateur. L'objectif étant de rendre le site web le plus agréable à lire possible, c'est à dire "responsive".
- **le dossier "scripts"** : La partie contenant tous les scripts JS. Encore une fois, les fichiers sont organisés en sous-dossier, avec un sous-dossier par page HTML. On retrouve aussi, à la racine de ce fichier, les scripts partagés, avec des fonctions communes à plusieurs fichiers. C'est ici aussi que l'on retrouve le sous-dossier "classes" qui contient les classes JS comme le gestionnaire de requêtes (RequestManager.js), ou le fichier Post.js, représentant la classe Post.
- **le dossier "php"** : La partie PHP. C'est ici que l'on retrouve les fichiers permettant de faire les requêtes à la base de données. Ce dossier peut sembler très imposant car il existe un fichier pour chaque requête, mais chacun des fichiers est en fait assez léger.
- **le dossier "res"** : La partie ressources, qui comprends nos images pour le site.

L'objectif de cet architecture est d'optimiser la répartition du code source du projet tout en améliorant sa lisibilité.

## 2.2 Base de données

La base de données est organisée en quatre tables, USER, POST, CATEGORY et APPLY.



### 2.2.1 USER

Chaque utilisateur est défini par les champs suivants :

- **id** : son ID, soit la clé primaire de la table,
- **pseudo** : un pseudo, qui doit être unique dans la table, car potentiellement utilisé pour la connexion,
- **mail** : un mail, qui doit aussi être unique car potentiellement utilisé pour la connexion,
- **color** : une couleur qui sera affectée aléatoirement à la création,
- **postalCode et city** : une adresse basique, constituée uniquement du code postal et de la ville, seul ces informations nous ont paru nécessaires à stocker,
- **isAdmin** : un booléen pour connaître les droits de l'utilisateur (true=admin, false!=admin),
- **password** : le mot de passe de l'utilisateur, stocké dans la table de façon cryptée, après avoir hashé et salé.

USER								
id	pseudo	mail	color	money	postalCode	city	isAdmin	password
1	Doe	doe@gmail.com	#4FB7FF	100	42	Saint-Etienne	0	\$2y\$10\$uJllyjz#HJyCUiXJucKM.uaeOdvUAehyokKUIIDoepp.xRnnvq@bwW

### 2.2.2 POST

Chaque post est défini par les champs suivants :

- **id** : son ID, soit la clé primaire de la table,
- **idOwner** : l'ID de son propriétaire, une référence à la table USER,
- **idCategory** : l'ID de sa catégorie, une référence à la table CATEGORY,
- **datePublication, startDate, endDate** : des chaînes de caractères pour les dates de publication du post, de début de la mission et de fin de la mission proposée par le post,
- **postalCode et city** : une adresse basique, constituée uniquement du code postal et de la ville d'exécution du service,

- **title et description** : un titre et une description du service demandé,
- **price** : le prix en étoiles proposé par l'utilisateur pour ce service.

POST										
id	idOwner	idCategory	datePublication	title	startDate	endDate	postalCode	city	description	price
5	1	2	12/05/2021	Cherche programmeur	13/05/2021	20/05/2021	69	Lyon	Bonjour, cherche programmeur dans le langage C pour mon projet	10

### 2.2.3 CATEGORY et APPLY

Les deux dernières tables ont pour but de stocker respectivement la liste des candidatures par service et les noms des catégories. La table APPLY a pour clé primaire les deux colonnes qui la constitue : idPost et idOwner

CATEGORY	
id	name
2	Informatique

APPLY	
idPost	idApplier
5	1

## 3 Déroulé du projet

### 3.1 Partage du travail

Pour un projet de cette taille, il est nécessaire de partager le travail à l'avance. Cela implique une séparation du projet en des parties distinctes et le plus indépendantes possible, permettant ainsi de travailler séparément sans devoir prendre un temps trop important lors de l'assemblage des travaux respectifs.

Au commencement du projet, nous avons pu identifier les différentes parties suivantes :

- **partie HTML/CSS** : création des pages HTML et CSS,
- **partie PHP** : requêtes à la base de données,
- **partie JS** : lien entre le HTML et le PHP

Nous avons créé ensemble les pages HTML et CSS, commençant la création doucement, avec pour objectif la conservation d'une cohérence au fil des pages. Nous n'avons pas défini à l'avance qui ferait quelles pages, nous avons préféré plutôt nous consulter après chaque fin de page pour en discuter et savoir s'il fallait reprendre certains éléments de la page ou commencer à en créer une nouvelle. Le premier s'est occupé de l'index, l'inscription, la connexion et le formulaire de demande de service, le second de la partie utilisateur avec la page d'accueil utilisateur, la gestion des profils pour l'administrateur, la gestion des services postés et le formulaire permettant de modifier le profil.

Il a fallu ensuite partager les deux parties restantes, à savoir le PHP et le JS. C'est l'aisance sur ces langages informatiques qui a permis de définir qui ferait quoi. Antoine s'est chargé de la partie JS car très à l'aise sur ce langage et Idriss de la partie PHP, nécessitant plus de recherche car aucun de nous n'avions eu l'occasion de travailler sur des projets personnels avec ce langage par le passé.

Pendant que nous avançons sur nos parties respectives, nous nous sommes entraides, tour à tour. Une fois la partie PHP finie, nous avons travaillé ensemble sur le JS pour finir de lier les informations de la base de données à la page HTML.

## **3.2 Difficultés rencontrées**

### **3.2.1 Rôle de l'administrateur**

La première difficulté que nous avons rencontré à été de définir le rôle d'un administrateur ainsi que les limites de ce dernier. Peut-il modifier une annonce ? La supprimer ? Est-ce utile d'inclure un bannissement des utilisateurs ? Devons-nous créer un super-administrateur ?

Nous avons finalement tranché. Il nous a semblé utile pour un administrateur de supprimer une annonce si celle-ci ne respecte pas la loi par exemple. En revanche, nous n'avons pas voulu laisser le droit aux administrateurs de modifier une annonce, cette dernière appartient à son créateur. Dans le même souci de respect des législations, il nous a paru essentiel de permettre à l'administrateur de bannir les utilisateurs non respectueux des règles de vie. Enfin l'administrateur sera la personne en charge de définir les autres administrateurs. La question s'est posée d'un super-administrateur auquel nous avons finalement décidé de ne pas répondre favorablement. L'ambition de ce site est d'avoir confiance en ses utilisateurs, ainsi seul deux niveaux de droits ont été créés : administrateur et simple utilisateur.

### **3.2.2 Publication du projet sur Webetu**

Lors du rendu de notre projet, il a été demandé d'ajouter la base de données sur webetu. Nous avons finalement compris la manière de procéder. Mais lors du rajout de nos fichiers sources, nous avons rencontré un problème majeur. Lors des requêtes XML, au lieu de lire le fichier PHP nous retournant une réponse correcte, la réponse retournée était directement le contenu du fichier PHP. Nous n'avons pas trouvé à temps de solution pour réparer ce problème.

### **3.2.3 Problèmes en phase de développement**

#### **3.2.3.a Lien entre PHP et JS**

Il a été compliqué dans un premier temps de connecter le JS avec le PHP. Pour se faire, nous avons décidé d'utiliser des requêtes XML HTTP. La donnée transite ainsi entre la requête SQL en PHP, la fonction JS et la page HTML sous la forme d'un objet JSON dont les champs sont similaires aux tables de notre base de données.

#### **3.2.3.b Redirection de pages**

Au démarrage de la session, l'utilisateur arrive normalement sur la page index.html. L'accès aux autres pages passe par une inscription puis une connexion. Cependant, si une personne se décidait à taper l'url direct de notre page "account-home.html" par exemple (page dédiée à la gestion du compte personnel), rien ne l'empêcherait d'y arriver.

C'est pourquoi il nous a semblé nécessaire de forcer la redirection vers la page `index.html` si la variable session n'est pas définie. Le temps de requête étant plus long que le temps de réponse des fichiers JS, nous avons décidé d'utiliser la session JS avec la fonction `sessionStorage`.

## 4 Job'Stone : une révolution

### 4.1 L'architecture, le retour

En prêtant attention à l'architecture des fichiers, on peut remarquer le même type de patterns : un dossier contient les fichiers d'un langage (HTML/CSS/JS/PHP), ce dossier est divisé en sous-dossiers (un pour chaque page), puis ce sous-dossier est divisé en fichiers utiles à la page. Cependant, lorsqu'on regarde plus attentivement, on remarque des fichiers à la racine de chaque dossier associé à un langage : ces fichiers sont des fichiers partagés. Le but a été de généraliser le plus possible les fonctions pour qu'elles soient partagées entre plusieurs fichiers, évitant ainsi la redondance des informations et des fonctions.

### 4.2 Direction : Les étoiles

La question du mode de paiement s'est posée au début du projet. Une forte volonté de se démarquer sur le marché et de proposer une monnaie alternative, sans conséquence sur la vie réelle, et prônant de justes valeurs d'égalité nous ont amené à choisir des étoiles.

Chaque utilisateur fait son entrée chez Job'Stone avec un montant offert de 1000 Star'Jobs, un montant considérable, même si la valeur de cette monnaie virtuelle n'est régie que par ses utilisateurs.

### 4.3 Vers la perfection grâce aux filtres

Job'Stone a vocation à devenir LA plateforme d'entraide entre particuliers. C'est pourquoi il est nécessaire de mettre en place des filtres permettant à l'utilisateur de se retrouver dans une multitudes d'annonces, pour trouver celle qui lui correspond. C'est pour cette raison que nous avons mis en place sur la page d'accueil un filtre sur la catégorie ainsi que sur la ville. Différent d'un filtre classique, notre filtre cherche en utilisant la méthode `LIKE`, permettant d'optimiser la recherche et ainsi la satisfaction utilisateur. Notre objectif aurait été d'implémenter une distance de Levenshtein permettant de proposer une liste d'éléments similaires à l'utilisateur.

Dans la page de gestion des membres (accessible uniquement aux administrateurs), une barre de recherche a aussi été ajoutée pour simplifier la recherche.

### 4.4 Le destin filtrera

Après avoir travaillé sur les filtres des catégories et des villes, nous avons réfléchi à un potentiel filtre sur les titres des postes ou des description de ces derniers. Nous n'avons pas retenue cette idée finalement. Inspiré des grands groupes de marketing de nos jours, nous avons pu remarquer que le fait de réduire les possibilités de filtrage augmentait les ventes du groupe, à condition que l'utilisateur soit apte à filtrer à minima les éléments attirant son attention. C'est pourquoi nous n'avons pas mis en place ce type de filtre.



## 4.5 PHP tu changeras, rapide tu seras

Afin de faciliter la portabilité et la maintenance de la plateforme avec le temps, nous avons jugé essentiel d'utiliser un fichier "PDO.php" pour déclarer la variable pdo utilisée dans nos fichiers PHP. Ainsi, il suffit de ne changer qu'un seul fichier, qu'une seule ligne pour automatiquement changer tous les accès à la base de données.

## 4.6 Administration

Un administrateur est capable de supprimer une annonce (l'option "Supprimer" se rajoute sur les postes de la pages d'accueil de ce dernier). Il peut aussi supprimer des utilisateurs (à travers le portail "Gestion utilisateur") ou bien gérer les autres administrateurs. En effet, chaque administrateur a le pouvoir de supprimer un autre administrateur, ou même de rabaisser un administrateur au rang de simple utilisateur (et inversement). Tout le monde ne doit pas être administrateur, seul les personnes vraiment motivées et dignes de confiance mériteront d'évoluer au sein de la plateforme.

## 4.7 Pseudo ou mail, c'est du pareil au même

Toujours dans un esprit d'améliorer l'expérience utilisateur, nous avons laissé la possibilité à l'utilisateur d'entrer son pseudo ou son mail comme identifiant à la connexion.

# 5 Conseils pour découvrir la plateforme

Pour naviguer au sein de la plateforme, nous vous proposons de d'abord tester l'intégralité des fonctionnalités en temps que simple utilisateur. Ensuite vous pourrez vous connecter avec le pseudo (ou mail) administrateur suivant : (id)admin (mail)admin@admin.com (password)admin

# 6 Conclusion

Ce projet est l'aboutissement de nos deux années de licence informatique. De l'architecture des fichiers jusqu'aux codes sources des fonctions, en passant par la gestion d'une base de données, nous pouvons dire que ce projet, en plus d'être intéressant, a été très enrichissant.

En plus d'être déjà pleinement fonctionnelle, cette plateforme pourrait être déployé dans un véritable but non-lucratif. Nous sommes heureux de pouvoir vous présenter aujourd'hui le résultat de notre travail et de notre investissement.