

EPREUVE E4

Conception et maintenance de solutions informatiques



Projet Personnel Encadré 3 Freelance Platform

BTS SIO OPTION SOLUTION LOGICIELLES APPLICATION METIERS

BABEL Yoken : yoken.babel@gmail.com

MONTIAS Timothée : timontias@gmail.com

SOMMAIRE

SOMMAIRE

LETTRE DE MISSION

CAHIER DES CHARGES

- Contexte et définition du problème

- Objectifs

- Contraintes

- Techniques

- Hébergement/Accessibilité

- Graphique

- Programmation

- Fonctionnelles

- Utilisateurs

- Compétences

- Sécurité

- Mots de passe

- Stabilité

- Confidentialité

- Description Fonctionnelles

- Technologies

- Logiciels

- Programmation

- Matériels/Fournisseur

- Améliorations Possibles

- Définitions

DOCUMENTATION TECHNIQUE

- Arborescence

- Fonctionnement

- Routes d'accès

- Ajout

- Modification

- Suppression

- Initialisation

NOTICE D'UTILISATION

- Généralité

- Projets

- Home

- Login/Logout

- Freelancer

- Mes Offres

- Ajout

- Modification

- Suppression

- Consulter les messages

- Information
 - Modification
 - CV
 - Photo de profile
 - Ajout
 - Modification
- Company
 - Projets
 - Modifier
 - Supprimer
 - Ajouter
 - Offres/Messages
 - Information
 - Modification
 - Photo de profile
 - Ajout
 - Modification
- Admin
 - Utilisateurs
 - Ajout
 - Modification
 - Suppression
 - Projets
 - Modification
 - Suppression
 - Message
 - Consultation
 - Suppression
- ANNEXES
 - MLD
 - Liens/Sources

1. LETTRE DE MISSION

Dans le cadre de votre BTS SIO option SLAM, vous devez réaliser un site internet permettant à des entreprises de déposer des projets et à des freelancers de répondre à ces projets.

Les projets devront être consultable par tous les utilisateurs et seul les freelancer connectés au site pourront postuler à ces projets.

Chaque entreprise pourra gérer ses projets et établir un contrat pour chaque projet avec un freelancer.

Les freelancers et les entreprises pourront discuter au sujet d'un projet mis en ligne

L'entreprise pourra avoir accès au profil de chaque freelancer qui postulera à ses projets notamment à son CV.

L'administrateur aura accès à la liste des utilisateurs, il pourra ajouter modifier ou supprimer un utilisateur. Il pourra faire de même avec les projets.

En ce qui concerne les messages échangés entre un freelancer et une entreprise au sujet d'un projet l'administrateur pourra consulter lesdits messages et supprimer la conversation si besoin.

Le site devra être programmé en langage interprété et posséder une base de données et être accessible en ligne.

2. CAHIER DES CHARGES

a. Contexte et définition du problème

Dans le cadre de notre formation BTS SIO (Services Informatiques aux Organisation) option SLAM (Solutions Logicielles Applications Métiers), nous avons dû effectuer des P.P.E (Projets Personnels Encadrés).

Ce cahier des a pour but de décrire le contexte ainsi que les modalités de production du PPE numéro 3 – Freelance Platform.

La société BelMont a été contactée par l'entreprise X afin de produire une plateforme de freelance permettant à des entreprises de déposer des projets et à des freelancers de répondre aux projets proposés sur la plateforme.

L'équipe produisant cette solution sera composée par :

- Timothée MONTIAS
- Yoken BABEL

En cas d'incompréhension des termes employés dans ce cahier des charges, vous pouvez vous référer au [Chapitre 7 : Définitions](#).

b. Objectifs

Ce projet a pour but de fournir une plateforme web pouvant accueillir différents projets. Ces projets seront uniquement destinés à des développeurs.

Cette plateforme contiendra un back-office pour l'administrer.

Voici une liste exhaustive des fonctions principales que devra remplir la plateforme :

- **Fonction principale 1** : Authentifier les utilisateurs
- **Fonction principale 2** : Gestion et visualisation des utilisateurs (Modification, Ajout et Suppression)
- **Fonction principale 3** : Gestion des droits d'accès des utilisateurs
- **Fonction principale 4** : Gestion et visualisation des projets (Modification, ajout et suppression)
- **Fonction principale 5** : Gestion et visualisation des messages (Suppression)
- **Fonction principale 6** : Sauvegarde et sécurisation des données
- **Fonction principale 7** : Création d'un contrat (validation d'une offre sur un projet par une entreprise)
- **Fonction principale 8** : Gestion et visualisation des offres (Modification, Ajout, Suppression)
- **Fonction secondaire 1** : Empêcher soumission d'une offre lorsqu'un contrat est édité pour le projet en question.
- **Fonction secondaire 2** : Empêcher un freelancer de faire plusieurs offres sur un seul et même projet.
- **Fonction secondaire 3** : Empêcher une entreprise de générer un second contrat pour le même projet.

c. Contraintes

Cette plateforme devra se plier à quelques contraintes que nous allons détailler ci-dessous

.

i. Techniques

1. Hébergement/Accessibilité

La plateforme devra tourner sur un serveur approprié. Elle devra être accessible à travers internet, où que ce soit en France (est ciblée une clientèle française).

Un Budget de 100 euros sera crédité pour héberger la plateforme ainsi que la base de données. Le choix de l'hébergeur ainsi que de la formule sera laissé à la liberté de la société BelMont qui livrera la solution fonctionnelle.

La base de données utilisera l'interface PHPMyAdmin comme moteur de gestion de base données.

2. Graphique

La plateforme devra être accessible sur tous types de terminaux, elle devra donc être adaptative. Pour cela la librairie Bootstrap est demandée et le choix du template sera laissé à la société BelMont.

Un budget de 50 euros sera attribué au choix du template

L'affichage de l'interface graphique du site internet sera géré par le moteur de template BLADE.

3. Programmation

Pour réaliser cette plateforme web un langage interprété est requis, c'est pourquoi le langage PHP est demandé. Nous recommandons une programmation orienté objet

Le [Framework](#) Laravel (version 5.3) est demandé pour la réalisation de cette plateforme web.

De cette manière la plateforme sera maintenable et plus simplement modifiable.

Pour la gestion des dépendances de la plateforme, Composer sera utilisé.

La gestion du travail collaboratif sera assuré par l'utilisation de Git ainsi que la plateforme GitHub.

ii. Fonctionnelles

1. Utilisateurs

Les utilisateurs auront un rôle spécifique qui leur fournira des droits d'accès à certaines fonctionnalités de la plateforme.

Voici la liste des rôles et de leurs droits d'accès aux fonctionnalités :

FREELANCER :

- Accès à la liste des projets mis en ligne
- Possibilité de postuler à un projet et de faire une offre
- Possibilité de mettre à jour ses informations personnelles
- Possibilité de gérer ses offres (Visualisation, Modification, Suppression)

COMPANY :

- Accès à la liste des projets mis en ligne (Impossible de postuler)
- Possibilité de gérer ses projets (Ajout, Visualisation, Modification, Suppression)
- Possibilité de mettre à jour ses informations
- Possibilité d'éditer un contrat avec un freelancer (Valider une offre)

ADMIN

- Accès à la liste des projets mis en ligne (via le site et via le back-office)
- Accès au back-office
- Possibilité de gérer les projets (Visualisation, Modification, Suppression)
- Accès à la liste des utilisateurs
- Possibilité de gérer les utilisateurs
- Possibilité de consulter les messages échangés entre une entreprise et un freelancer.
- Possibilité de Supprimer une conversation

Pour des schémas détaillés de rôles, rendez-vous dans le chapitre [Diagrammes](#)

2. Projets

Les projets seront enregistrés en base de données et seront consultables par tout utilisateurs se rendant sur le site, qu'il soit authentifié ou pas.

Une fois le projet réalisé par un freelancer il sera supprimé de la base de données et aucune archive ne sera conservée.

3. Offres

Les offres sont uniquement faites par les utilisateurs authentifiés en tant que freelancer.

Chaque freelancer pourra consulter la liste de ses offres et pourra modifier à tout moment une offre qu'il a faite.

Une fois l'offre faite une conversation commencera entre l'entreprise et le freelancer au sujet du projet.

Une offre pourra être supprimée si et seulement si elle n'a pas de contrat.

4. Contrats

Un contrat ne peut être généré uniquement par une entreprise suite à une offre déposée par un freelancer.

Une fois le contrat honoré et le projet livré et le freelancer payé, le projet, ainsi que le contrat, l'offre, la conversation associés seront supprimés de la base de données et aucune archive ne sera gardée.

iii. Sécurité

1. Mots de passe

Les mots de passe seront stockés dans la base de données (BDD).

De ce fait, les mots de passe devront être cryptés lors de leur insertion en BDD.

Le cryptage du mot de passe se fera par hashage et est géré automatiquement par Laravel. Ainsi il sera décrypté par le Framework Laravel lors de l'authentification de l'utilisateur en saisissant son nom d'utilisateur et son mot de passe.

2. Stabilité

Le niveau de stabilité doit être élevé, mais le système n'est pas critique. Il appartient donc à la société BelMont qui livrera la solution de prendre les mesures afin que l'accès à la plateforme soit toujours et en toute circonstance accessible via internet. Pour cela nous allouons une extension de budget de 50 euros.

3. Confidentialité

Etant donné le caractère personnel et confidentiel des données stockées en BDD, cette dernière devra faire l'objet d'une déclaration à la CNIL avant sa mise en production.

De plus tous les moyens nécessaires à la sécurisation des données personnelles doivent être employés afin de garantir le respect de la vie privée des utilisateurs de la plateforme.

De ce fait l'accès à certaines données sont limités en fonction du rôle de l'utilisateur, cet aspect sera géré par le Framework Laravel, ainsi qu'une authentification obligatoire pour utiliser pleinement la plateforme.

d. Description Fonctionnelle

Numéro	Libelle	Objectif	Priorité
1	Authentifier les utilisateurs	S'assurer de l'identité des utilisateurs qui se connectent à la plateforme	Haute
2	Gestion et visualisation des utilisateurs	Permettre la suppression, modification, l'ajout ainsi que l'affichage des utilisateurs	Haute
3	Gestion des droits d'accès des utilisateurs	S'assurer que les utilisateurs aient accès aux parties de l'application que leurs sont dédiées	Haute
4	Gestion et visualisation des projets	Permettre la suppression, modification, l'ajout ainsi que l'affichage des projets	Haute

5	Gestion et visualisation des offres	Permettre la suppression, modification, l'ajout ainsi que l'affichage des offres	Moyenne
6	Sauvegarde et sécurisation des données	S'assurer de la stabilité de la BDD et de la sécurisation des données personnelles des utilisateurs	Très Haute
7	Gestion des Contrat	Permettre la Modification, Ajout, suppression et visualisation des contrats.	Haute

e. Technologies

i. Logiciels

Pour le visionnage du code source de la plateforme, Git et le site web associé GitHub seront utilisés.

L'IDE (Interface de Développement) PhpStorm sera utilisée pour développer la solution.

ii. Programmation

La solution sera développée en PHP 5.6 avec le Framework Laravel 5.3

Les langages HTML 5 et CSS 3 seront utilisés pour l'interface graphique.
Le choix du template pour cette dernière revient à la société BelMont qui livrera la solution.

La base de données sera écrite en langage SQL sous le moteur InnoDB

iii. Matériels/fournisseurs

Le choix de la société BelMont s'est porté vers OVH pour héberger la plateforme.

f. Améliorations possibles

Amélioration n°1 : Générer un envoie de mail automatique lorsqu'un freelancer postule à un projet.

Amélioration n°2 : Générer un envoie de mail automatique lorsqu'une entreprise répond à un freelancer et vis-versa

Amélioration n°3 : Créer des tests à faire passer aux freelancer afin qu'ils valident des qualifications.

Amélioration n°4 : Générer un système de paiement via PayPal pour que l'entreprise paie le freelancer une fois la solution livrée

Amélioration n°5 : Créer un forum où les utilisateurs pourront échanger entre eux, qu'ils soient authentifiés ou non.

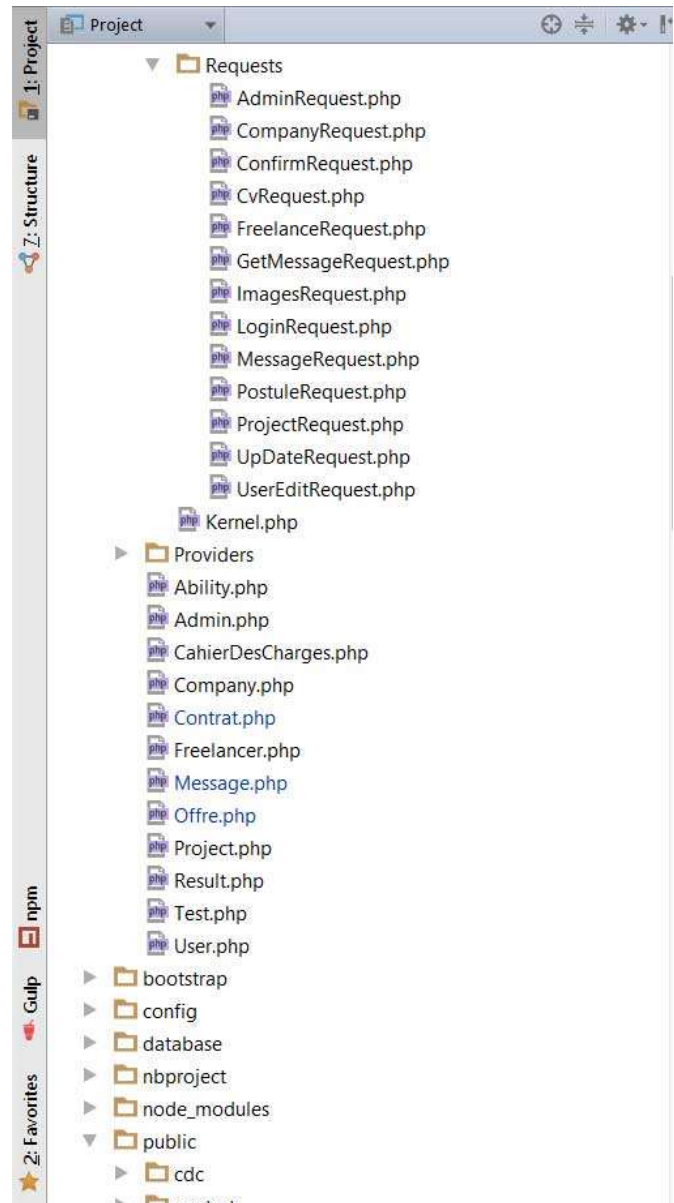
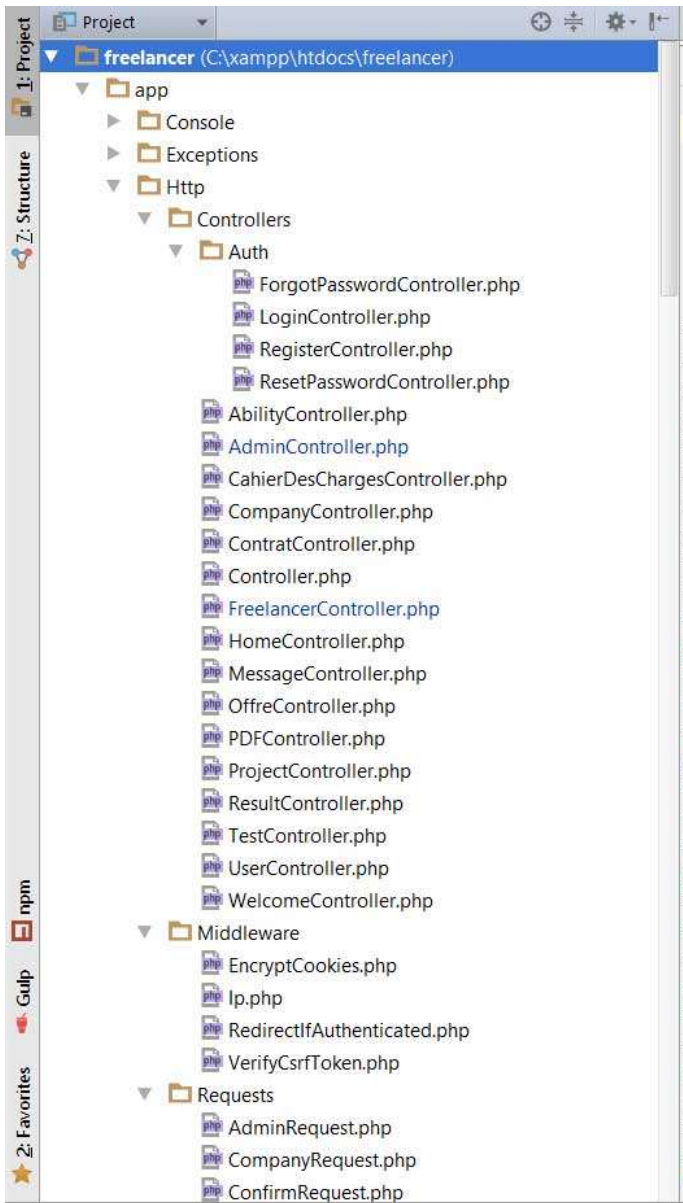
g. Définitions

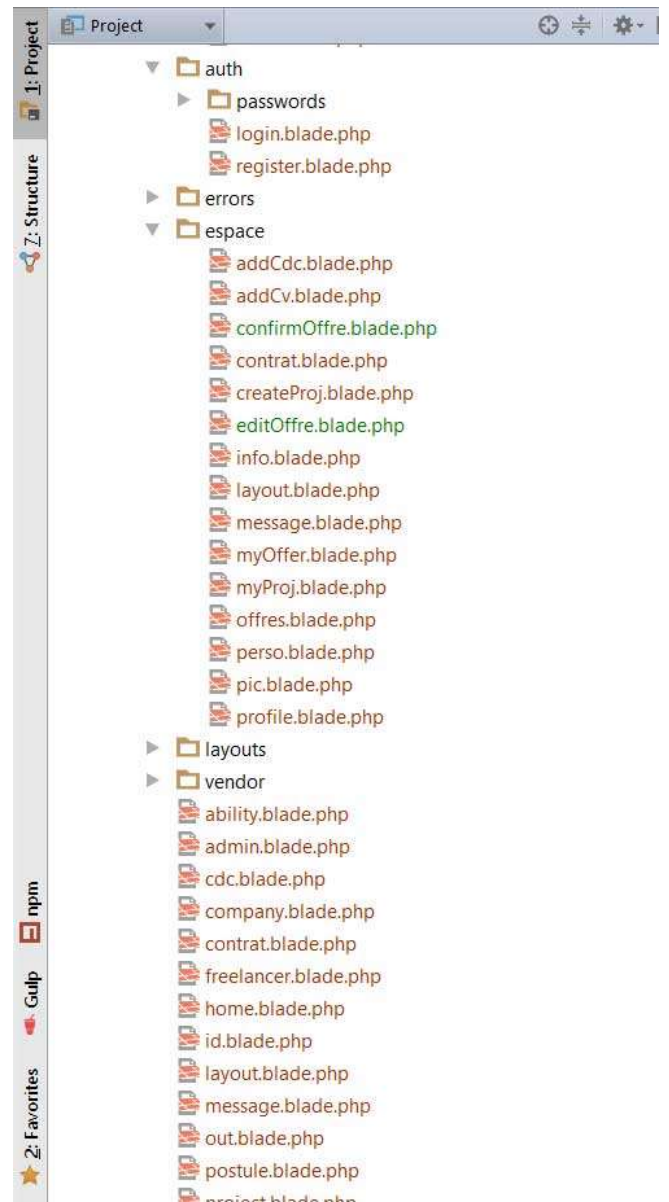
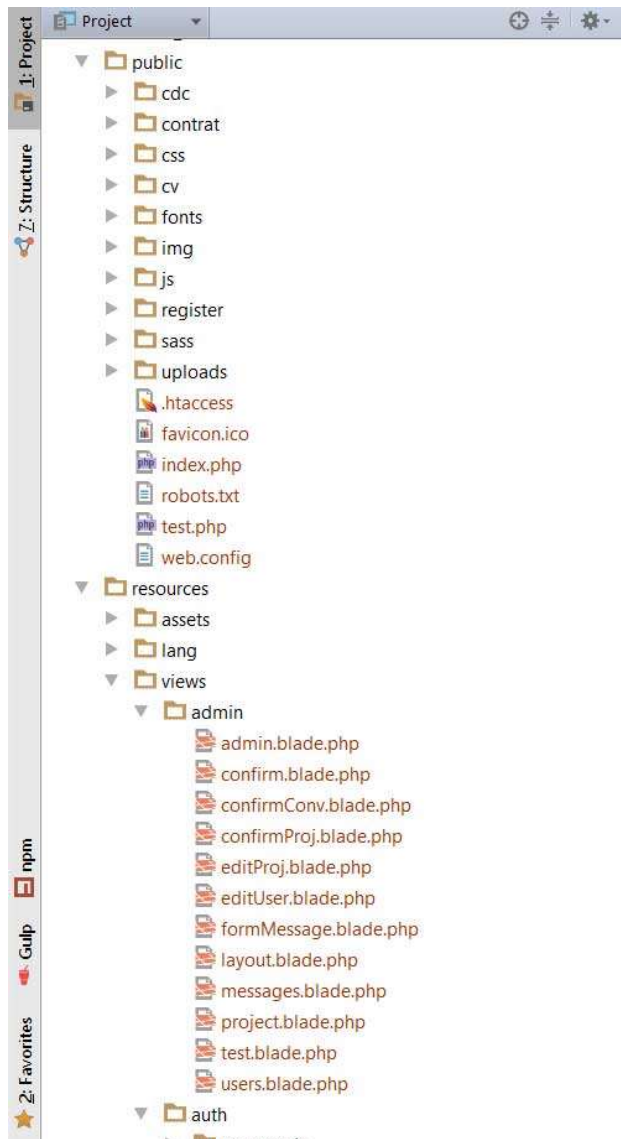
Framework : Collection de classe PHP permettant un développement plus complexe mais plus facile à maintenir et à debugger.

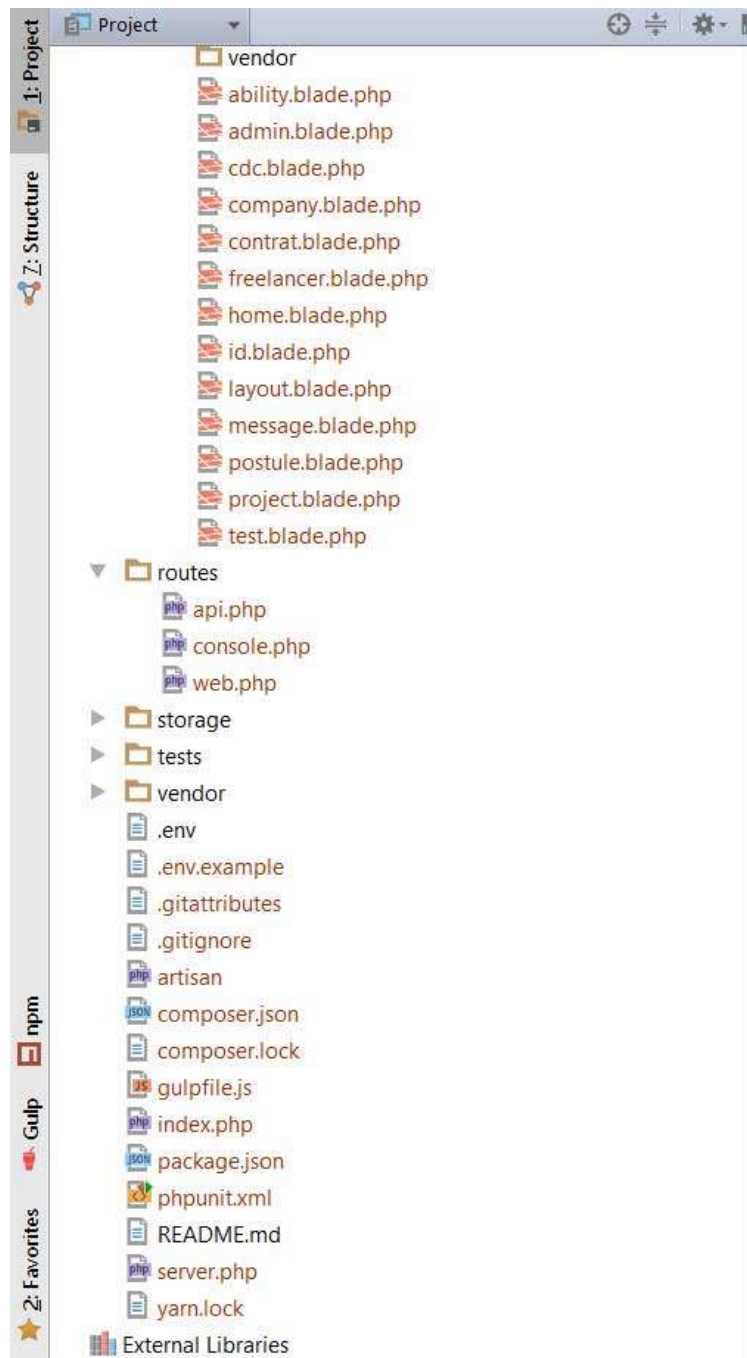
Versionnage : Fait de classer les différentes versions d'un fichier en fonction des modifications apportées à ce dernier. Un Logiciel de versionnage permettra de gérer les différentes versions et de pouvoir restaurer le fichier a une date antérieure.

3. DOCUMENTATION TECHNIQUE

a. Arborescence







b. Fonctionnement

i. Routes d'accès

Quand un utilisateur clique sur un lien de la plateforme, le chemin sur lequel il est redirigé est contenu dans le fichier [web.php](#). Ce fichier est contenue dans le dossier [routes](#) à la racine du projet (Exemple : Une série de route de la plateforme)

```
13 /*
14 *
15 *
16 * Cette route génère la page d'accueil.
17 * Lorsque l'url contient simplement '/'
18 * alors l'utilisateur est renvoyé vers le contrôleur WelcomeController
19 * et la fonction index de ce dernier est appelé.
20 * Cette fonction va renvoyer vers la vue home.php (page d'accueil)
21 */
22 */
23 Route::get('/', 'WelcomeController@index');
24
25 /*
26 *
27 * Cette route va appeler la fonction redirect du contrôleur WelcomeController
28 * Cette fonction renvoie la vue permettant de définir le type d'utilisateur cherchant à s'inscrire
29 *
30 */
31
32 Route::get('id', 'WelcomeController@redirect');
33
34
35 /*
36 *
37 * Ici nous avons un groupe de routes
38 * toutes les routes dont la racine est 'register'
39 * sont contenues dans ce groupe
40 *
41 */
42 */
43 Route::group(['prefix' => 'register'], function () {
44
45     /*
46     *
47     * Nous avons la deux route identique, une en get et un en post
48     * Le get va traiter l'url une fois que l'utilisateur aura cliqué sur un lien et donc génère l'url.
49     * Le post va traiter cette url une fois un formulaire valide.
50     * La route freelance permet donc d'enregistrer un freelance.
51     */
52 }
```

Chaque route appelle une fonction qui se trouve dans un fichier appelé controller. Les controllers se trouvent dans le dossier App/Http/Controllers

```
28
29 Route::get('users', 'AdminController@userList');
30
```

Cette route fera appel à la fonction userList() du controller AdminController.php

```

96
97 public function userList() {
98     $role = Auth::user();
99     if(Auth::check() == true) {
100         if($role->role == 'admin') {
101             $user = new User();
102             $users = $user->findAll();
103             return view('admin.users', compact('users'));
104         }
105         else
106             return 'Accès réservé a l\'administrateur';
107     }
108     else
109         return redirect('login');
110 }
111

```

La méthode (fonction) userList() va appeler une autre méthode de la classe User.php, la méthode findAll(), elle va permettre de trouver tous les utilisateurs contenue dans la base de données (BDD).

```

41
42 public function findAll() {
43     return DB::table('users')->select('*')->distinct()->get();
44 }

```

Ainsi le Contrôleur les transmettra à la vue [users.blade.php](#) dans le dossier [ressources/views/admin](#) et nous pourrons ainsi afficher les utilisateurs contenus en BDD

```

1 @extends('admin.layout')
2
3 @section('content')

```

Ce fichier est une extension du fichier layout.blade.php contenu dans le même dossier

ii. Ajout

Par Exemple pour l'ajout d'un projet, la route vers le formulaire d'ajout sera :

```
180 // Ces routes appellent les fonction createProject et storeProject du ProjectController
181 //Ces fonctions permettent à une entreprise de créer un projet
182 Route::get('create/project', 'ProjectController@createProject');
183 Route::post('create/project', 'ProjectController@storeProject');
```

Il y aura une route en get et un en post. La route en get fera la redirection vers le formulaire d'ajout d'un projet tandis que la route en post fera le traitement de ce formulaire sur la même route.

La route get va appeler la fonction createProject() dans ProjectController.php

```
98 public function createProject(){
99     if(Auth::check() == true){
100         $user = Auth::user();
101         if($user->role == 'company'){
102             return view('espace.createProj', compact('user'));
103         }
104         return redirect('/')->with('error', 'accès restreint');
105     }
106     return redirect('login')->with('error', 'veuillez vous authentifier');
107 }
108 }
```

Dans la méthode createProject() on vérifie que l'utilisateur est bien authentifié et qu'il s'agisse bien d'une entreprise.

Une fois les vérifications faites la méthode appelle la vue createProj.blade.php qui va nous afficher le formulaire à remplir pour créer le projet.

La fonction storeProject() sera appelée après validation du formulaire par la route en post vu sur la photo précédente.

```

109
110 public function storeProject(){
111
112     $cahier = new CahierDesCharges();
113     $proj = new Project();
114     $comp = new Company();
115
116
117
118     if(Auth::check() == true) {
119         $user = Auth::user();
120         if($user->role == 'company') {
121             $cuser = $comp->getByUserId($user->id);
122
123             $cahier->deadline = ProjectRequest::capture()->deadline;
124             $cahier->company_id = $cuser->id;
125             $cahier->url_odc = 'temp/'.$cuser->id;
126
127             $cahier->save();
128
129             $cdc = $cahier->getCdcByUrl('temp/'.$cuser->id);
130
131             $proj->title = ProjectRequest::capture()->title;
132             $proj->description = ProjectRequest::capture()->description;
133             $proj->price = ProjectRequest::capture()->price;
134             $proj->company_id = $cuser->id;
135             $proj->cdc_id = $cdc->id;
136
137             $proj->save();
138
139             return redirect('add/odc');
140         }
141         return redirect('/')->with('error', 'accès restreint');
142     }
143     return redirect('login')->with('error', 'veuillez vous authentifier');
144 }
145
146
147

```

Dans cette méthode on vérifie l'authentification de l'utilisateur et si il s'agit en effet d'une entreprise

Une fois les vérifications faites on instancie un nouveau cahier des charges, un nouveau projet et une nouvelle entreprise.

Une fois les objets instanciés on peut créer les objets avant de les enregistrer en base de données avec l'instruction `$mvariable->save()` ;

iii. Modification

Par exemple la modification d'un utilisateur, se déroule à peu près de la même manière que l'ajout à la différence que l'on envoie des données à modifier.

La route utilisée :

```
121 */
122 Route::get('users/{id}', 'AdminController@editUser');
123 Route::post('users/{id}', 'AdminController@postUser');
124
```

Il y a comme pour l'ajout 2 routes, une en post et une en get. La route en get comme pour l'ajout va nous afficher le formulaire à remplir pour modifier les informations de l'utilisateur sélectionné.

```
207
208 public function editUser($id) {
209     $connect = Auth::user();
210     if(Auth::check() == true){
211         if($connect->role == 'admin'){
212             $user = new User();
213             $theUser = $user->findById($id);
214             $usercomp = $theUser;
215             if($theUser->role == 'company'){
216                 $comp = new Company();
217                 $usercomp = $comp->getByUserId($theUser->id);
218             }
219             return view('admin.editUser', compact('theUser', 'usercomp'));
220         }
221         return 'accès réservé à l\'administrateur';
222     }
223     return redirect('login');
224 }
225
```

Dans cette méthode editUser() nous vérifions l'authentification et le rôle de l'utilisateur. Une fois les vérifications faites on instancie un nouvel objet User. Une fois l'objet instancié on récupère les informations de l'utilisateur à modifier grâce à la méthode findById() déclarée dans la class User().

```

53
54 public function findById($id){
55     return DB::table('users')->select('*')->where('id', $id)->first();
56 }
57

```

Cette méthode findById() trouve un utilisateur en fonction de son id (l'id est unique pour chaque utilisateur)

Une fois récupérer on vérifie que le rôle de l'utilisateur à modifier est une entreprise, si c'est la cas il y aura des données supplémentaire à récupérer donc on instancie un objet company et on récupère les données grâce à la méthode getByUserId() déclarée dans la classe Company.php

```

37
38 public function getByUserId($id){
39     return DB::table('company')->select('*')->where('user_id', $id)->first();
40 }

```

Une fois toutes les informations récupérées le controller les envoie à la vue et génère un formulaire prérempli avec les informations stockées en BDD.

L'administrateur n'aura plus qu'à modifier les champs qu'il souhaite modifier et valider le formulaire.

La validation du formulaire va appeler la route en post que l'on a vue plus haut.

La route en post va appeler la méthode postUser() :

```

226
227 public function postUser($id){
228     $connected = Auth::user();
229     if(Auth::check() == true){
230         if($connected->role == 'admin'){
231             $find = new User();
232             $user = $find->findById($id);
233             if($user->role == 'company'){
234                 try{
235                     $comp = new Company();
236                     $comp->updateByUserId($user->id, UserEditRequest::capture()->sirot, UserEditRequest::capture()->denomination);
237                 }
238                 catch(Exception $e){
239                     return redirect('admin')->with('error', $e);
240                 }
241             }
242             try{
243                 $find->updateUserBy($user->id, UserEditRequest::capture()->name, UserEditRequest::capture()->firstname, UserEditRequest::capture()->lastname);
244                 return redirect('admin')->with('success', 'Utilisateur modifié correctement');
245             }
246             catch(Exception $e){
247                 return redirect('admin')->with('error', $e);
248             }
249         }
250         return 'Accès réservé à l\'administrateur';
251     }
252     return redirect('login');
253 }
254

```

Dans cette méthode on vérifie l'authentification et le rôle de l'utilisateur authentifié.

Une fois fait on instancie les objets afin de pouvoir modifier les données.

Les données seront modifiées par les fonctions update des classes User et Company (Company uniquement si l'utilisateur modifié est une entreprise).

```

58 public function upDateUserById($id, $name, $firstname, $username, $mail, $address, $phone){
59     return DB::table('users')
60         ->where('id', $id)
61         ->update([
62             'name' => $name,
63             'username' => $username,
64             'firstname' => $firstname,
65             'mail' => $mail,
66             'address' => $address,
67             'phone' => $phone,
68         ]);
69 }

```

Cette méthode ci-dessus est la méthode déclarée dans la classe User et elle est identique à la méthode upDateUserById déclarée dans la classe Company.

iv. Suppression

Par exemple la suppression d'une offre.

La suppression comme l'ajout et la modification se fait en deux étapes.

Une route en get et une en post.

La route en get va nous donner un formulaire de confirmation par sécurité.

```

194 // Ces routes permettent de supprimer une offre d'un freelancer en appelant les fonctions
195 // confirmDeleteOffreMess et DeleteOffreAndMessage du FreelancerController
196 Route::get('confirm/delete/{f_id}/{p_id}/{c_id}', 'FreelancerController@confirmDeleteOffreMess');
197 Route::post('confirm/delete/{f_id}/{p_id}/{c_id}', 'FreelancerController@DeleteOffreAndMessage');
198 );

```

La méthode confirmDeleteOffreMess() va retourner le formulaire de confirmation. Un menu déroulant avec deux choix « oui » ou « non »

```

264 public function confirmDeleteOffreMess(){
265     if(Auth::check() == true){
266         $user = Auth::user();
267         if($user->role == 'freelancer'){
268             return view('espace.confirmOffre', compact('user'));
269         }
270         return redirect('perso')->with('error', 'Accès restreint');
271     }
272     return redirect('login')->with('error', 'veuillez vous authentifier');
273 }

```

La méthode DeleteOffreMessage() sera appelée une fois le formulaire validé
La valeur du formulaire sera vérifiée. Si l'utilisateur a mis « oui » alors les données seront supprimées de la BDD


```

274
275 public function DeleteOffreAndMessage($f_id, $p_id, $c_id){
276     if(Auth::check() == true){
277         $user = Auth::user();
278         if($user->role == 'freelancer'){
279             if(ConfirmRequest::capture()->confirm == 'oui'){
280                 $offre = new Offre();
281                 $off = $offre->findByForeignKey($f_id, $p_id);
282
283                 $contrat = new Contrat();
284                 $cont = $contrat->findByForeignKeys($f_id, $c_id, $off->id, $p_id);
285
286                 if($cont == ''){
287                     $offre->deleteByForeign($f_id, $p_id);
288                     $message = new Message();
289                     $message->deleteByUsersAndProjId($f_id, $p_id, $c_id);
290                     return redirect('perso')->with('success', 'L\'offre à bien été supprimée');
291                 }
292                 return redirect('perso/offres')->with('error', 'Un contrat est établie sur cette offre impossible de la supprimer sans
293
294             }
295         }
296         return redirect('perso/offres');
297     }
298     return redirect('perso')->with('error', 'Accès restreint');
299 }
300 return redirect('login')->with('error', 'veuillez vous authentifier');
301 }
302
303
304

```

Si un contrat est attaché à l'offre elle ne pourra pas être supprimée.
Plusieurs objets seront instanciés et plusieurs requêtes faites pour trouver quelles données supprimer.

```

83
84 public function deleteByForeign($f_id, $p_id){
85     return DB::table('offre')
86         ->where('freelancer_id', $f_id)
87         ->where('project_id', $p_id)
88         ->delete();
89 }
90
91 }

```

Cette méthode supprime en fonction des clefs étrangères. En effet il est important de faire attention aux relations des tables lorsqu'une modification est apportée à la BDD que ce soit ajout, modification ou suppression.

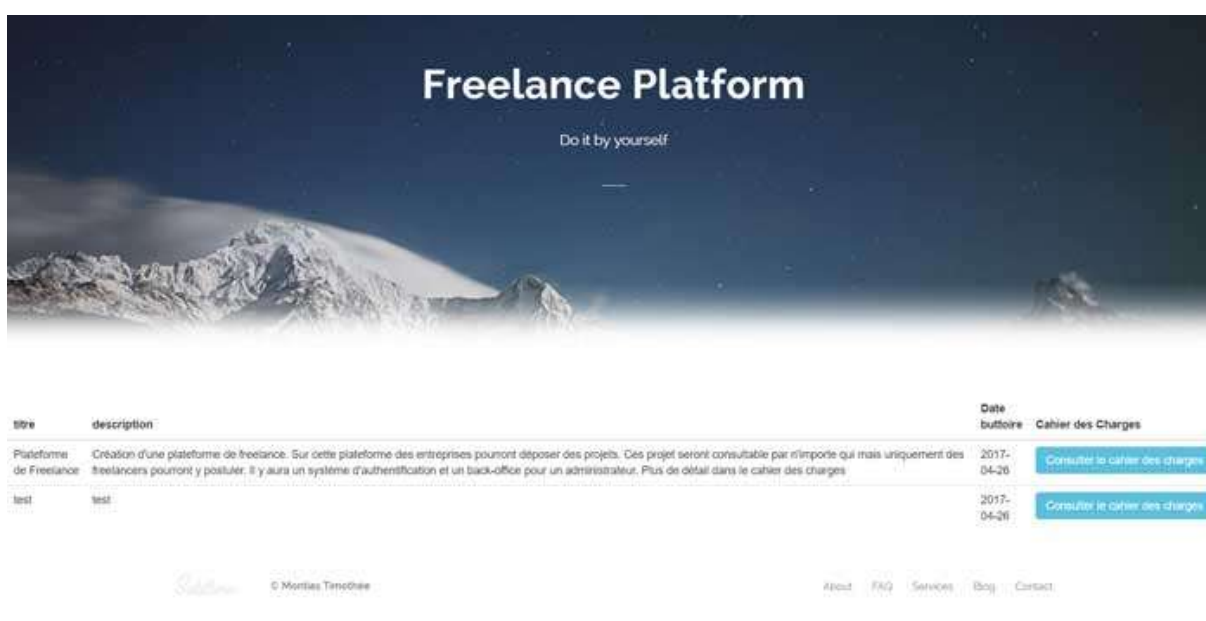
4. NOTICE D'UTILISATION

a. Généralités

En arrivant sur le site l'utilisateur sera sur la page d'accueil et il aura alors la possibilité de choisir dans le menu différents liens.



i. Projets



En cliquant sur le liens projets l'utilisateur aura la possibilité de découvrir les projets qui ont été déposé par les entreprises.

Si l'utilisateur est authentifié en tant que freelancer il aura alors des boutons « postuler » qui apparaitront en marge des projets



i. Home

Le lien de menu Home renvoie sur la page d'accueil de la plateforme.

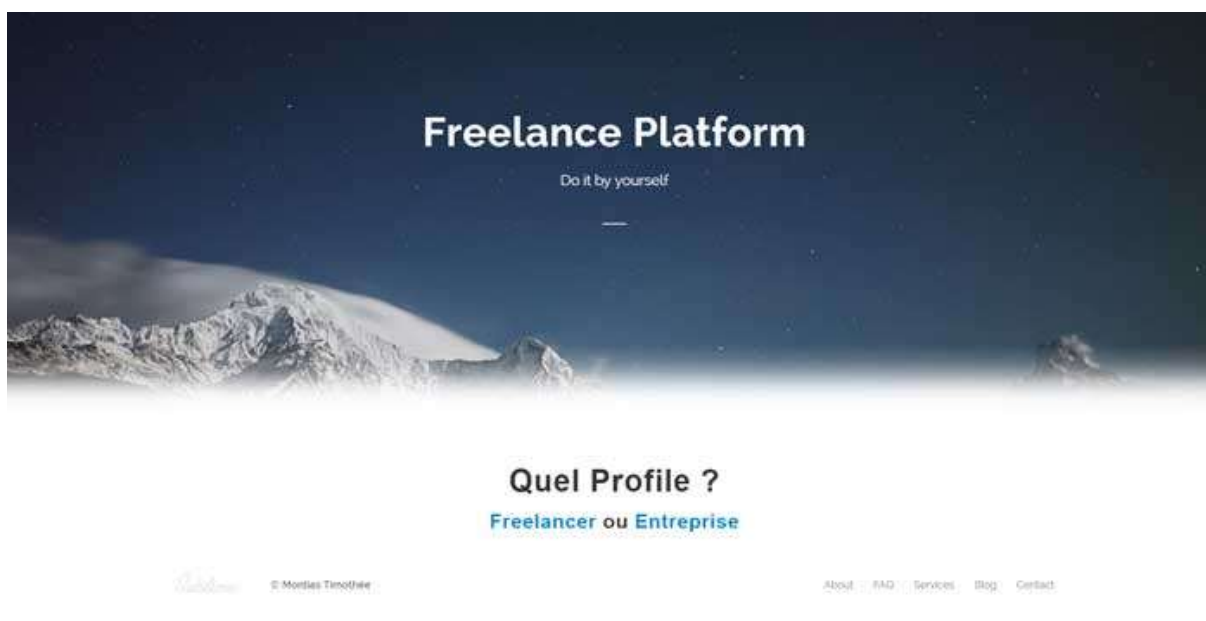
ii. Login/Logout

Le lien Login permet à un utilisateur de s'authentifier sur le site afin de déposer des projets s'il s'agit d'une entreprise ou de déposer des offres s'il s'agit d'un freelancer.

The image shows a login form with a light blue header containing the word "Login". Below the header, there are two input fields: the first is labeled "Nom d'utilisateur" and the second is labeled "Votre mot de passe". At the bottom of the form, there are two buttons: a blue button labeled "Login" on the left and a blue button labeled "Nouveau ?" on the right.

En sélectionnant le lien login on arrive sur ce formulaire d'authentification. Si l'internaute ne possède pas de compte il a la possibilité d'en créer un en cliquant sur le bouton « Nouveau ? »

Si l'internaute créer un compte il arrivera sur une page intermédiaire :



Il sera demandé à l'internaute de choisir un profil.

Pour le profil Entreprise il aura ce formulaire ci-contre

Un freelancer aura le même formulaire avec cependant 2 champs en moins ceux renseignant le SIRET et la Dénomination

Une fois créer le nouveau profil, l'utilisateur sera redirigé vers la page login.

Il n'aura alors plus qu'à s'authentifier.

Une fois authentifier il sera redirigé vers son espace personnelle et le menu subira alors quelques changements

Pour une entreprise le liens « Mes Offres » sera transformé en « Mes Projets »

Inscription:

Nom du représentant légal:
Nom

Prénom de représentant légal:
Prénom

Nom d'utilisateur:
Pseudo

Mail :
example@example.com

Mot de passe :
mot de passe

Confirmation mot de passe :
confirmation

Téléphone :
0606060606

Dénomination :
SAS

SIRET :
12345678912345

Adresse :
53 chemin du puit

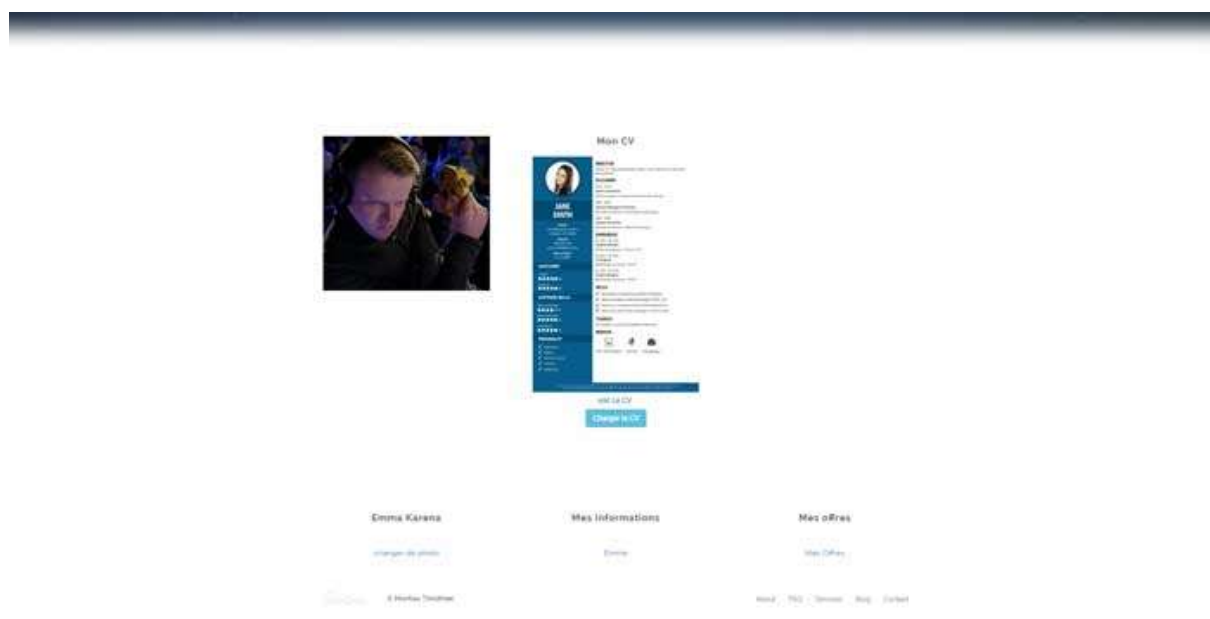
Code Postal :
11230

Ville :
Le Pays

Inscription

b. Freelancer

Une fois authentifié le freelancer se retrouve dirigé vers son espace personnel :



Il aura alors plusieurs choix, Il pourra consulter la liste des projets avec le pouvoir de postuler, de consulter ses offres, la possibilité de choisir une photo de profil comme avatar, la possibilité de choisir un CV qui sera visible pas les entreprises auprès desquelles il postule et enfin il aura la possibilité de modifier ses informations personnelles

i. Mes Offres

Intitulé du projet	Cahier des charges	Description du projet	Date buttoire	Message d'offre
Plateforme de Freelancee	consulter le cahier des charges	Création d'une plateforme de freelance. Sur cette plateforme des entreprises pourront déposer des projets. Ces projet seront consultable par n'importe qui mais uniquement des freelancers pourront y postuler. Il y aura un système d'authentification et un back-office pour un administrateur. Plus de détail dans le cahier des charges	2017-04-28	Bonjour, Je pourrais produire la solution demandée dans le temps imparti pour 900 euros. Cordialement

1. Ajout

Le bouton postuler, qui apparaît à droite en marge des projets dans la liste des projets permet à un freelancer authentifié d'ajouter une offre en base de donnée

Après avoir cliqué sur le bouton postuler correspondant à l'offre à laquelle il s'intéresse le freelancer est dirigé vers un formulaire afin de soumettre son offre.



Postuler

Message

Postuler

2. Modification

Pour modifier l'offre il suffit de cliquer sur le bouton avec le crayon et le freelancer sera dirigé vers un formulaire prérempli pour modifier l'offre comme il le désire.

Modifier mon offre

message

Bonjour,
Je pourrais produire la solution demandée
dans le temps impartis pour 900 euros.
Cordialement

Envoyer !

3. Suppression

Pour supprimer une offre il suffit de cliquer sur la croix à côté de l'offre en question.

Toutefois il sera impossible de supprimer une offre ayant un contrat.

Une fois cliqué sur la croix le freelancer est dirigé vers un formulaire de confirmation pour une mesure de sécurité.



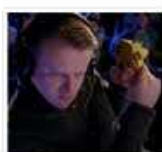
A confirmation dialog box with a light blue header containing the text "Voulez-vous supprimer l'offre sélectionnée ?". Below the header is a white input field with a dropdown arrow on the right, currently displaying the word "oui". To the right of the input field is a blue button with the text "Confirmer".

Si l'offre n'a pas de contrat elle sera supprimée.

4. Consulter les messages

Pour chaque offre faites par le freelancer il a la possibilité de voir si l'entreprise lui a répondu puisque une conversation entre les deux entités est engagé dès le moment où le freelancer postule à un projet.

Pour cela il lui suffit de cliquer sur le bouton « consulter les messages » comme indiqué sur la photo plus haut.



Emma Karena emma@karena.fr

Bonjour, Je pourrais produire la solution demandée dans le temps imparti pour 900 euros. Cordialement Emma Karena



Jean-Claude Duss jean-claude@duss.fr

Je ne suis pas intéressé par votre profil désolé Bien cordialement



Jean-Claude Duss jean-claude@duss.fr

Finalement j'accepte votre offre et vous valide un contrat

Poster un Message

Ecrire votre message

Poster le message

Il arrivera alors sur cette page et pourra converser avec l'entreprise au sujet du projet.

ii. Informations

Le freelancer a la possibilité de modifier ses informations personnelles via son espace personnel. Pour cela sur la page d'accueil de son espace personnel il suffit de cliquer sur son prénom sous le titre Mes Informations.

Une fois cliqué sur le lien le freelancer sera dirigé vers un formulaire prérempli lui permettant de modifier ses informations.

Modifier mes informations

Prénom

Nom

Pseudo

Email

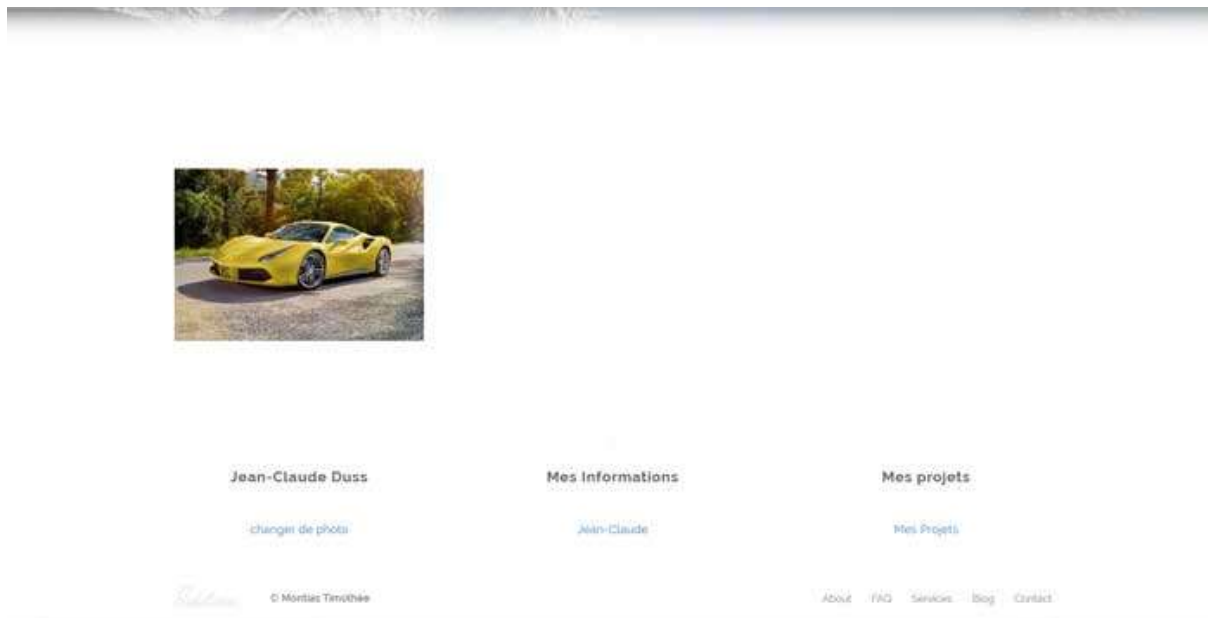
Adresse

Téléphone

[changer de mot de passe](#) [Envoyer !](#)

c. Company

Une fois authentifié l'entreprise sera dirigé vers son espace personnel qui se rapproche de celui du freelancer.



L'entreprise a alors le choix de consulter la liste de ses projets ou alors d'aller voir ses informations personnelles.

i. Projets

En sélectionnant Mes Projets l'entreprise est dirigée vers la liste de ses projets. Projets qui ne sont en cours.

Bonjour Jean-Claude

[Nouveau projet](#)

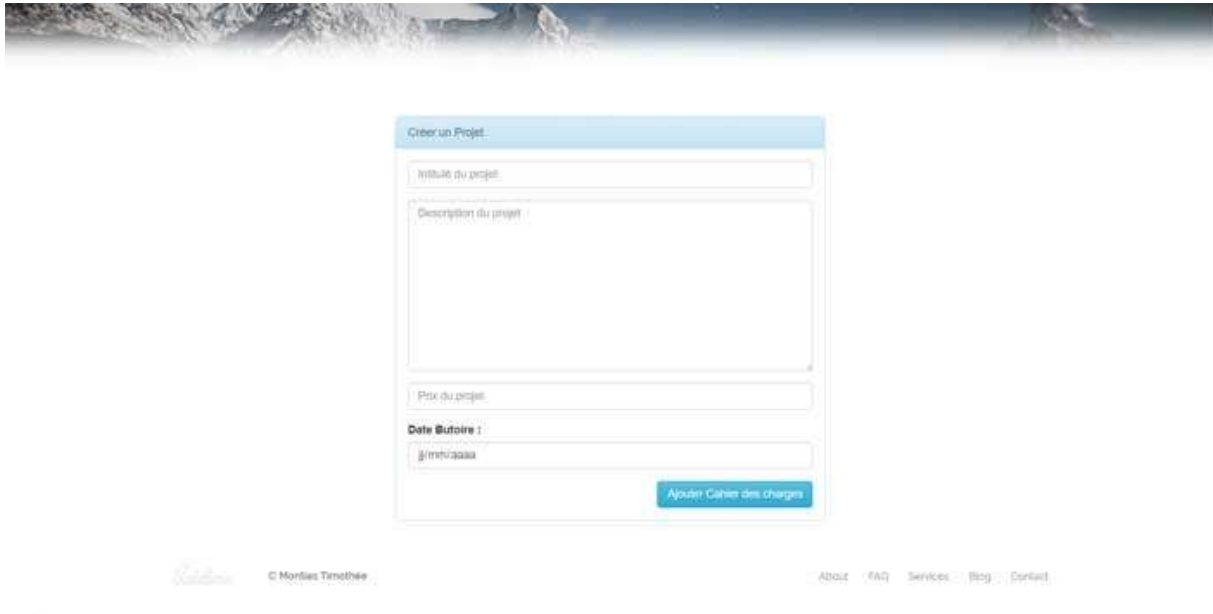
Intitulé	Description	Prix	Date butoir	Cahier des Charges	Offre
Plateforme de Freelance	Création d'une plateforme de freelance. Sur cette plateforme des entreprises pourront déposer des projets. Ces projet seront consultable par n'importe qui mais uniquement des freelancers pourront y postuler. Il y aura un système d'authentification et un back-office pour un administrateur. Plus de détail dans le cahier des charges	1500	2017-04-26	Cahier des charges	Consulter les Offres
test	test	1500	2017-04-26	Cahier des charges	Consulter les Offres

[About](#) [FAQ](#) [Services](#) [Blog](#) [Contact](#)

L'entreprise a alors la possibilité de créer un nouveau projet ou de consulter les offres qui ont été faite sur le projet déposé.

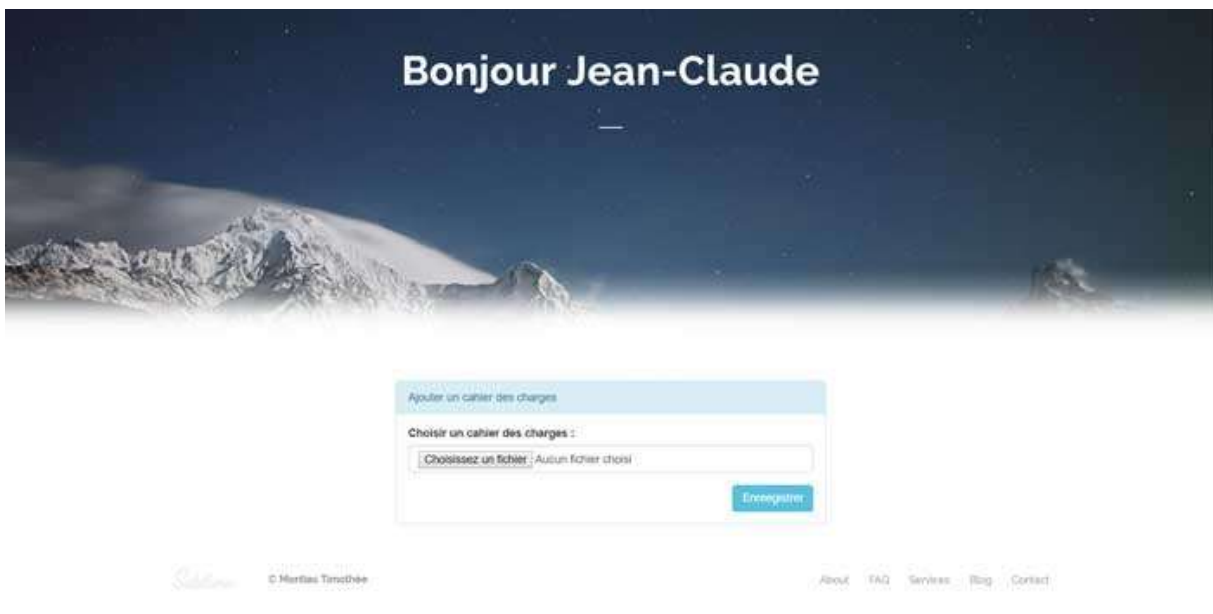
1. Ajout

Lorsque l'entreprise clique sur le bouton nouveau projet, elle est dirigée vers un formulaire lui permettant de créer un nouveau projet.



The screenshot shows a web page with a mountain background. A central form titled "Créer un Projet" contains the following fields: "Intitulé du projet:", "Description du projet:" (a large text area), "Prix du projet:", "Date Butoire:" (with a date picker showing "jj/mm/aaaa"), and a blue "Ajouter Cahier des charges" button. The footer includes a logo, "© Montias Timothée", and navigation links: "About", "FAQ", "Services", "Blog", and "Contact".

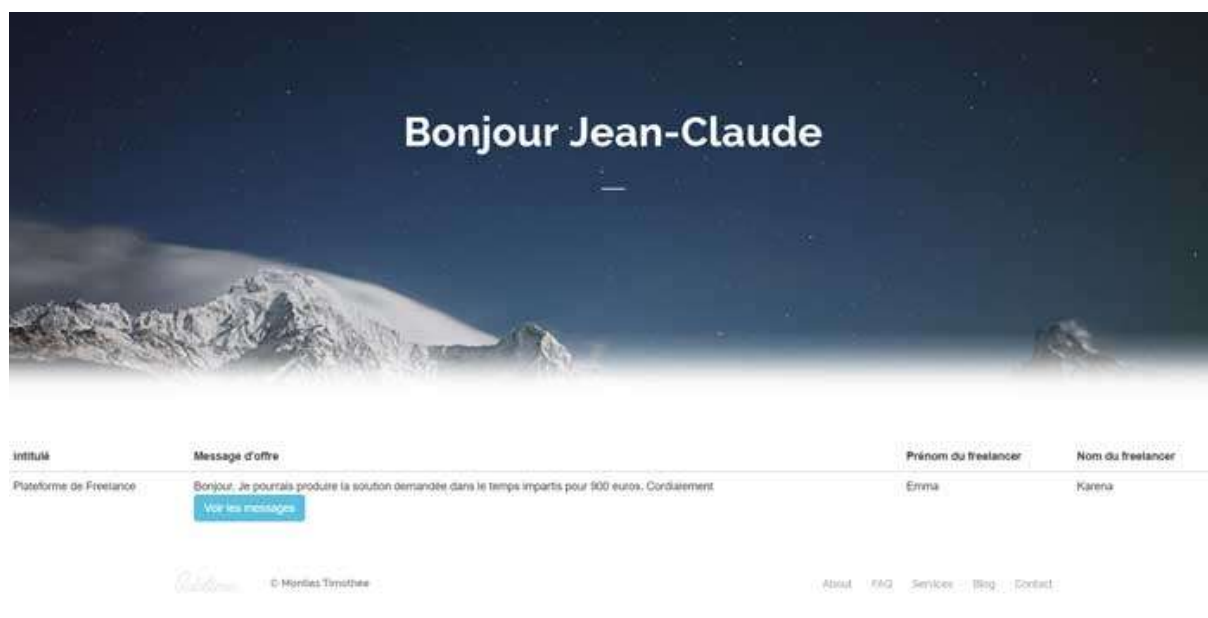
Une fois les champs renseignés, il est demandé de choisir un cahier des charges



The screenshot shows the same website background. A central form titled "Ajouter un cahier des charges" contains the following fields: "Choisir un cahier des charges:" (a dropdown menu showing "Choisissez un fichier" and "Aucun fichier choisi"), and a blue "Enregistrer" button. The footer is identical to the previous screenshot, showing the logo, "© Montias Timothée", and navigation links: "About", "FAQ", "Services", "Blog", and "Contact".

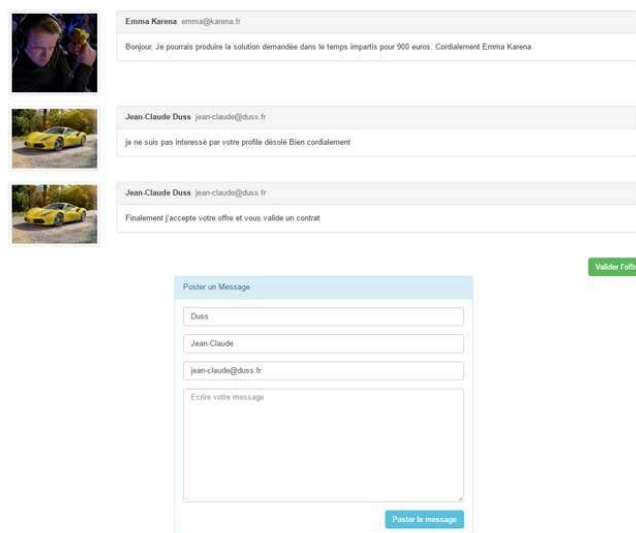
2. Offres/Messages

Lorsque l'entreprise cliquera sur le bouton consulter les offres, la liste des offres faites par différents freelancer s'affichera selon le projet sélectionné.



Et pour chaque offre il y a la possibilité de consulter les messages échangés entre l'entreprise dépositaire du projet et le freelancer postulant.

L'entreprise a la possibilité de générer un contrat si l'offre lui convient. Il aura un formulaire lui demandant d'uploader un fichier qui sera le contrat.



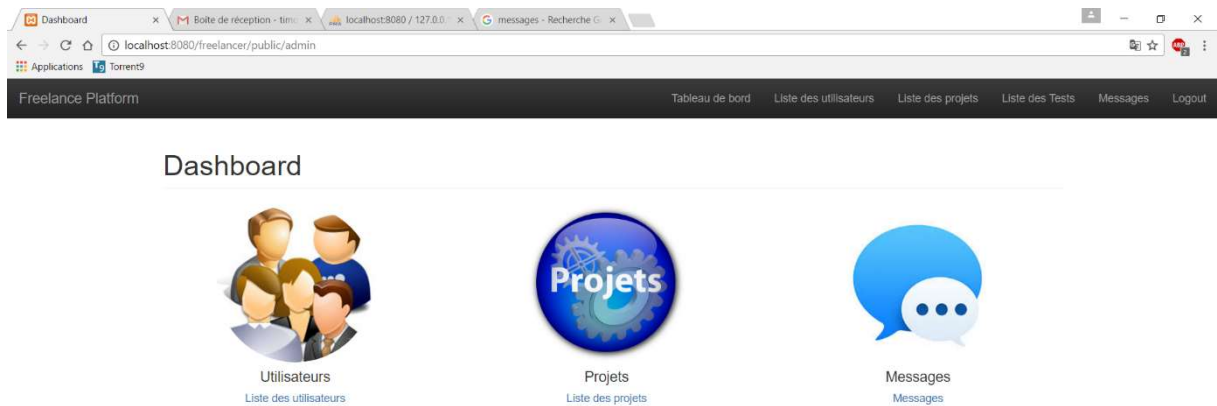
ii. Informations

Comme pour le freelancer si l'entreprise clique sur son prénom sous le titre « Mes Informations » elle sera dirigée vers un formulaire prérempli avec ses informations personnelles qu'il pourra modifier.

Pour avoir un aspect graphique veuillez, vous référer à la photo dans la partie freelancer sous partis Informations.

d. Admin

Une fois authentifié l'administrateur est dirigé vers son tableau de bord :



Une fois sur son tableau de bord, il a plusieurs choix.

Il peut soit gérer les utilisateurs, soit consulter, modifier ou supprimer les projets ou alors aller consulter les conversations entre freelance et entreprise

i. Utilisateurs

Lorsque l'administrateur clique sur le liens « liste des utilisateurs » ladite liste lui est affichées avec de nouvelles possibilités.

The screenshot shows a web browser window with the URL `localhost:8080/freelancer/public/admin/users`. The page title is "Users List". At the top, there are four tabs: "Freelancer" (selected), "Company", "All", and "New". Below the tabs is a table with the following columns: "Prénom", "Nom", "Nom D'utilisateur", "Email", "Adresse", "Téléphone", and "Catégorie". The table contains four rows of user data. Each row has two action buttons on the right: a yellow pencil icon for editing and a red 'X' icon for deleting.

Prénom	Nom	Nom D'utilisateur	Email	Adresse	Téléphone	Catégorie
Emma	Karena	freelance	emma@karena.fr	5 allée des chataignes 15480 L'Envale	622708556	freelancer
Jean-Claude	Duss	company	jean-claude@duss.fr	4 allée du robinet 75200 Paris	615264524	company
Timothée	Montias	admin	timontias@gmail.com	5 rue du docteur Dalbera 06390 Contes	622708555	admin
Undeuxttest	Testundeux	unttest	test@test.test	56 allée des platanes 12520 La manche	622708555	freelancer

Il a le choix d'organiser l'affichage selon les catégories.

Il peut aussi ajouter un nouvel utilisateur, ou alors en modifier ou supprimer un ou plusieurs.

1. Ajouter

En ce qui concerne l'ajout d'un utilisateur cela revient exactement au même lorsqu'un utilisateur s'inscrit pour la première fois, donc pour voir la marche à suivre merci de vous référer à la section Généralités Login/Logout.

2. Modifier

Lorsque l'administrateur clique sur modifier il est dirigé vers un formulaire prérempli contenant les informations de l'utilisateur sélectionner et il peut le modifier.

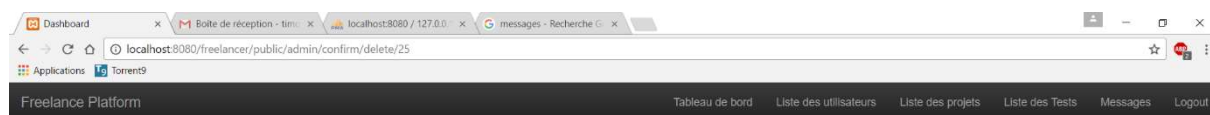
The screenshot shows a web browser window with the address bar displaying 'localhost:8080/freelancer/public/admin/users/25'. The browser has several tabs open: 'Dashboard', 'Boîte de réception - tim...', 'localhost:8080 / 127.0.0.1', and 'messages - Recherche'. The page title is 'Freelance Platform'. The navigation bar includes links for 'Tableau de bord', 'Liste des utilisateurs', 'Liste des projets', 'Liste des Tests', 'Messages', and 'Logout'. The main content area displays a form titled 'Modifier mes informations' with the following fields:

- Prénom**: Emma
- Nom**: Karena
- Pseudo**: freelance
- Email**: emma@karena.fr
- Adresse**: 5 allée des chataignes 15480 L'Envale
- Téléphone**: 622708556

An 'Envoyer !' button is located at the bottom right of the form.

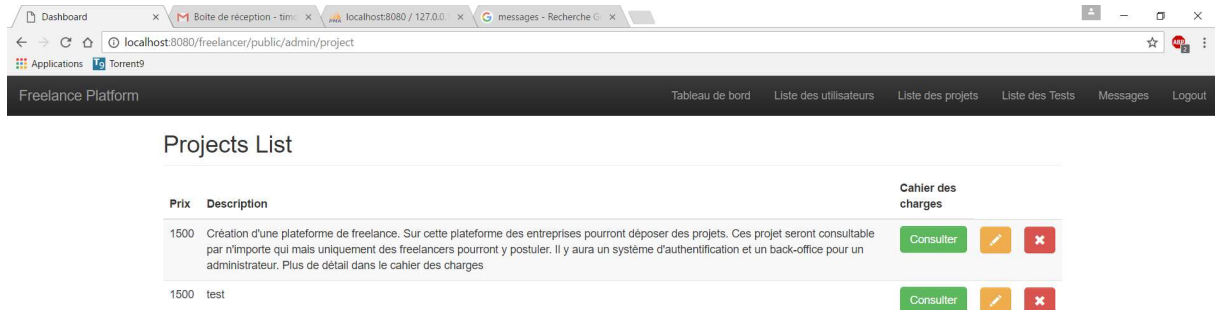
3. Supprimer

Si l'administrateur choisit de supprimer un utilisateur, par mesure de sécurité il sera dirigé vers une page de confirmation pour la suppression de l'utilisateur.



ii. Projets

Si l'administrateur choisit le lien « liste des projets » c'est cette dernière qui viendra à s'afficher.



Il peut alors soit modifier, soit supprimer le projet.

1. Modifier

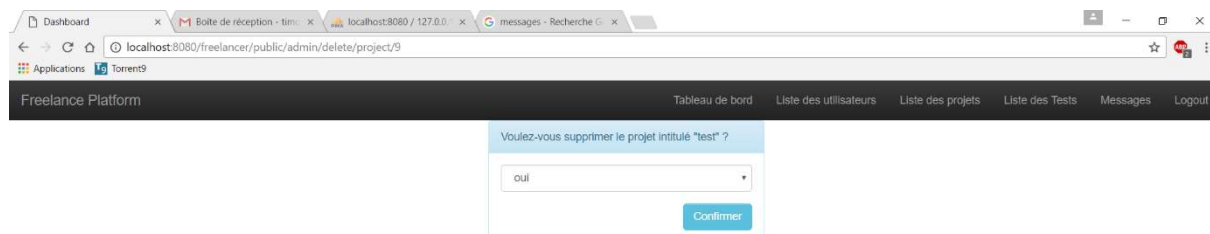
Lorsque l'administrateur choisit de modifier un projet il est là encore dirigé vers un formulaire prérempli pour modifier le projet il a aussi un champs textarea supplémentaire lui permettant de notifier l'entreprise du changement

The screenshot shows a web browser window with the URL `localhost:8080/freelancer/public/admin/edit/project/8`. The page title is 'Freelance Platform'. The navigation bar includes links: 'Tableau de bord', 'Liste des utilisateurs', 'Liste des projets', 'Liste des Tests', 'Messages', and 'Logout'. The main content area is titled 'Modifier Le projet' and contains the following form fields:

- Intitulé du projet**: A text input field containing 'Plateforme de Freelance'.
- description du projet**: A text area containing the text: 'Création d'une plateforme de freelance. Sur cette plateforme des entreprises pourront déposer des projets. Ces projet seront consultable par n'importe qui mais uniquement des freelancers pourront y postuler. Il y aura un système d'authentification et un back-office pour un administrateur. Plus de détail dans le cahier des charges'.
- Prix du projet**: A text input field containing '1500'.
- date buttoire**: A text input field containing '26/04/2017'.
- Message pour l'entreprise**: A large text area for sending a message to the company.
- Envoyer**: A blue button to submit the form.

2. Supprimer

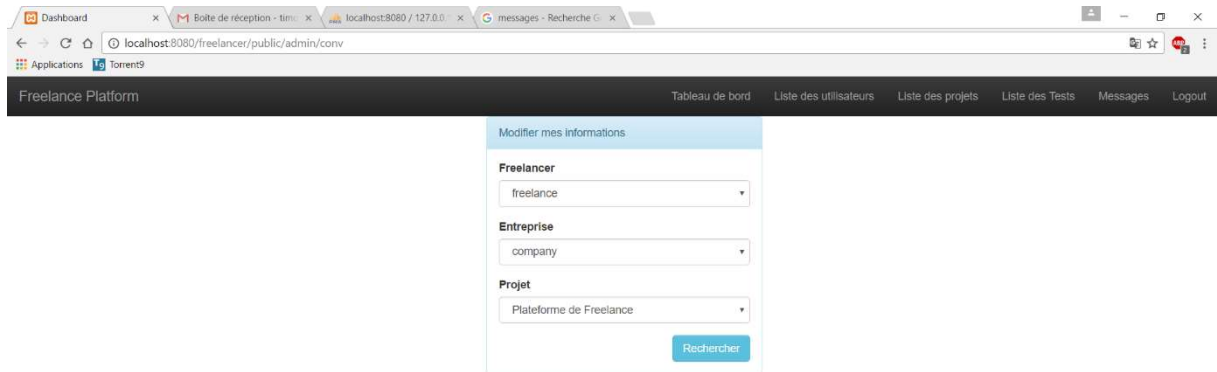
Lorsque l'administrateur choisit de supprimer un projet il est dirigé vers un formulaire de confirmation par mesure de sécurité



iii. Messages

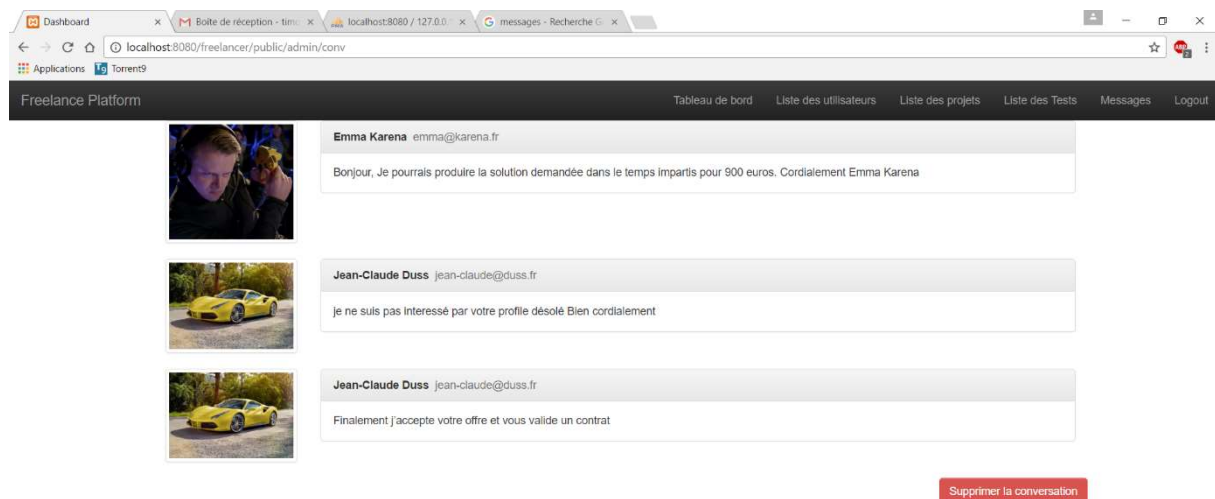
1. Consulter/Supprimer

En cliquant sur le lien « Messages » l'administrateur est dirigé vers un formulaire afin de saisir les paramètres de la conversation qu'il veut lire. A savoir les interlocuteurs par pseudo et le projet sujet de la discussion



The screenshot shows a web browser window with the URL `localhost:8080/freelancer/public/admin/conv`. The page title is "Freelance Platform". The navigation bar includes links: "Tableau de bord", "Liste des utilisateurs", "Liste des projets", "Liste des Tests", "Messages", and "Logout". The main content area displays a form titled "Modifier mes informations" with three dropdown menus: "Freelancer" (selected: "freelance"), "Entreprise" (selected: "company"), and "Projet" (selected: "Plateforme de Freelance"). A blue "Rechercher" button is at the bottom right of the form.

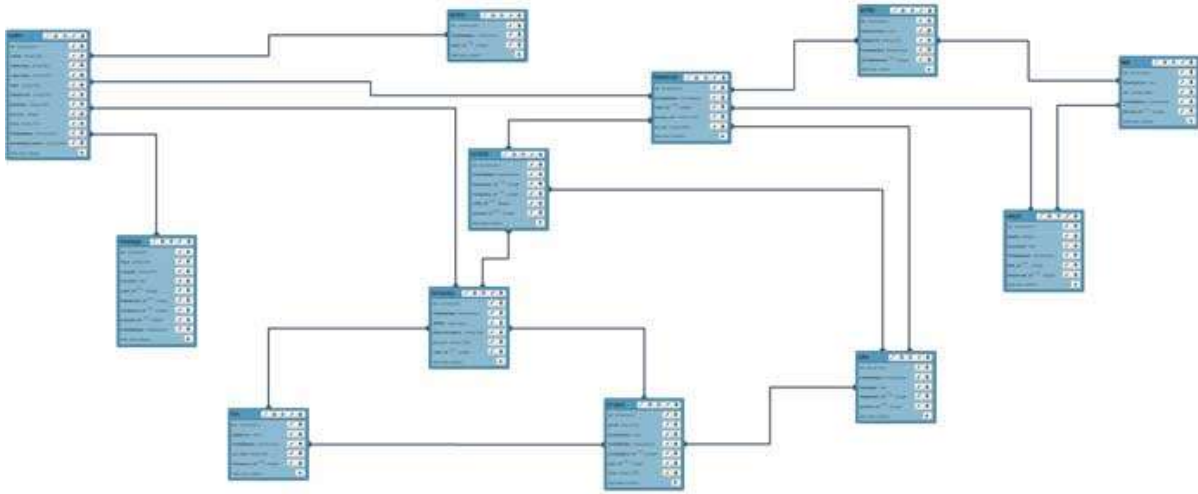
Une fois les paramètres saisis, l'administrateur peut lire la conversation, si les paramètres saisis ne mènent à aucune conversation il sera redirigé vers le formulaire.



L'administrateur a alors la possibilité de supprimer la conversation et si il clique sur le bouton, supprimer la conversation il est alors dirigé vers le formulaire de confirmation

5. ANNEXES

a. MLD



b. Liens/Sources

<http://www.laravelsd.com/5853e578887c/templates>

<https://laravel.com/>

<https://openclassrooms.com/courses/decouvrez-le-framework-php-laravel>

<http://laravel.sillo.org/>

<https://laracasts.com/>

<https://www.grafikart.fr/formations/laravel>