



# Rapport de stage

Danois Timéo

Dans La tête dans la toile

Avec Alan Ferronnière,

Du 4 Janvier 2024 au 8 Février 2024

## Table des matières

Table des matières	
Remerciement .....	3
Introduction .....	4
Les premiers jours en tant que stagiaire .....	5
La première réunion avec le théâtre .....	6
Création des comptes utilisateurs .....	8
Création des covoiturages .....	11
Création de la liste des trajets .....	13
Formulaire d'ajout de covoiturage .....	15
Interface de détails d'un trajet.....	17
1 - Pour une personne qui recherche un covoiturage .....	17
2 - Pour la personne qui propose le trajet.....	18
3 - Pour un employé des laboratoires vivants .....	20
Développement du système de message et des emails	
.....	22
1 - Le stockage des messages.....	22
2 - Affichage et enregistrement des messages.....	23
3 - Création des emails et de la fonction pour les envoyer. ....	24
Création de la page de nos covoiturages.....	26
Mise en place des messages administrables .....	27
Conclusion.....	28
Annexes.....	29

## Remerciement

Je tiens tout d'abord à exprimer ma sincère gratitude envers mon maître de stage, Alan Ferronnière pour m'avoir accueilli et permis de travailler dans son entreprise La Tête Dans La Toile. J'ai ainsi pu mettre en pratique et approfondir mes connaissances.

Merci également au théâtre Francine Vasse pour leur accueil et leur bienveillance. J'ai trouvé leur projet de covoiturage pour leur site très intéressant.

Enfin je souhaite adresser des remerciements à EPSI pour nous avoir prêté leurs locaux durant ces cinq semaines.

## Introduction

J'ai eu la chance de passer cinq semaines de stage dans l'entreprise La tête dans la toile avec Alan Ferronnière mon maître de stage. Il a créé et dirige cette agence de développement. L'entreprise développe des sites web sur mesure pour ses différents clients (particuliers ou entreprises).

Durant ce stage j'ai travaillé avec Rémi Jegard un autre stagiaire. Nous avons travaillé sur le site internet « les laboratoires vivants » qui est le site de la salle de théâtre Francine Vasse à Nantes. Alan a développé leur site depuis plusieurs années.

La salle avait comme projet de rajouter un système de covoiturage et copiétonnage directement sur leur site. Ça a donc été notre mission de stage avec Rémi. Nous devions programmer le système de création des covoiturages par des utilisateurs, mais aussi l'affichage et la réservation de ceci. Pour cela nous avons donc dû créer un système de comptes pour les utilisateurs, ainsi qu'un moyen de discussion entre eux.

Durant le stage nous faisions 35h par semaines, je travaillais du lundi au vendredi. Les horaires n'étaient pas fixes, c'est-à-dire que nous pouvions gérer notre temps nous même à condition que notre travail avance correctement et qu'à la fin de la semaine nous ayons nos 35h. Deux à trois jours dans la semaine étaient en distanciels et le reste en présentiel. Les jours en distanciel nous parlions entre stagiaires pour travailler, se partager le travail et débattre des façons de développer. Et si nous avions des problèmes qu'entre stagiaires nous n'arrivions pas à résoudre, nous faisons appel à l'expertise d'Allan. Lors des journées en présentiel, nous montrions les avancées du développement et nous recevions les nouvelles directives pour le site.

Le site est géré avec wordpress, ainsi nous développons en php avec un peu de javascript et en utilisant des extensions wordpress, comme ACF, CPT UI, Loco translate, Ultiamte Member, mais vous allez le découvrir dans la partie suivante qui est le déroulement du stage.

## Les premiers jours en tant que stagiaire

Le stage a commencé le jeudi 4 janvier. Comme nous n'avions jamais utilisé Wordpress, les deux premiers jours nous avons effectué des recherches pour apprendre son fonctionnement. Donc WordPress est un système gratuit qui permet de créer des sites Internet même sans connaissances en programmation. Il fonctionne avec des plugins et des thèmes qui permettent d'ajouter des fonctionnalités au site, certains sont payants et d'autres gratuits. Il utilise le système de gestion de base de données MySQL. La base de données permet de stocker tous les éléments du site (pages, articles, utilisateurs, etc.) dans des tables. Pour être clair, c'est un logiciel qui permet de créer des sites basiques sans avoir une grande connaissance en informatique, et on peut améliorer ces sites en modifiant directement le code du site et en utilisant des extensions.

Après avoir fait quelques tests de notre côté, le lundi nous avons récupéré les données du site « les laboratoires vivants » sur notre PC, grâce à FileZilla. Ainsi nous avons chacun une version locale du site sur nos machines. Cela permet donc de développer chacun de notre côté sans impacter le développement des autres ou le site en production. Pour mettre en commun nos avancées, nous avons utilisé Git qui permet de travailler à plusieurs sur un projet. On peut, avec, s'envoyer, comparer et récupérer du code, tout en gardant l'historique de nos versions précédentes du code.

Donc pendant deux jours nous avons parcouru le code et le WordPress du site pour se familiariser avec et réfléchir aux méthodes que nous allions devoir mettre en place. Nous connaissions la mission que nous allions devoir accomplir, mais nous avons attendu de rencontrer le théâtre, le mercredi, avant de commencer le développement du système de covoiturage. Ainsi nous allions échanger de vive voix avec le client afin d'être sûrs d'avoir toutes les informations concernant ce qu'ils souhaitaient mettre en place.

Si vous souhaitez voir à quoi ressemble le site pour vous faire une idée, il vous suffit de taper « les laboratoires vivants », ou de rechercher directement sur internet via ce lien : <https://www.leslaboratoiresvivants.com/>

## La première réunion avec le théâtre

Le projet de rajouter une partie covoiturage et copiétonnage sur la page internet du théâtre était géré par Adèle Locq, chargée de communication et billetterie et Angélique Bougeard, directrice de production. Nous les avons rencontrées au début du stage avec Remi Jegard et Alan Ferronnière pour discuter des objectifs du stage.

Avant la réunion nous avons reçu comme exemple le site de l'aéronef de Lille qui propose un système de covoiturage semblable à la demande. Nous avons donc listé les options de leur site, ainsi que le fonctionnement de la création d'un covoiturage, la réservation etc. Pendant la réunion nous avons discuté de toutes ces options pour savoir si certaines d'entre elles devaient être sur le système de covoiturage des Laboratoires vivants. Alan, Adèle et Angélique avaient déjà discuté au par avant de ce sujet, ils avaient donc déjà des idées qu'ils nous ont exposées.

Pour commencer on nous a demandé de développer juste le covoiturage sans se préoccuper du copiétonnage dans un premier temps. Cependant nous avons quand même travaillé en pensant qu'après le site devait pouvoir marcher avec ce système. Nous devions donc créer plusieurs choses :

Premièrement, nous devons concevoir un formulaire pour créer un covoiturage. Quand un utilisateur veut créer un covoiturage, il doit entrer le Type de trajet (aller, retour, aller-retour), il doit dire pour quel événement ainsi que la date de celui-ci. Il donne l'heure de départ et l'heure d'arrivée. Le nombre de places disponibles. Et il doit écrire une description du trajet avec les étapes, les points d'intérêts, etc. Cette description va permettre aux autres utilisateurs de voir si le conducteur passe à un endroit intéressant pour eux. Afin d'ajouter une sécurité et de ne pas avoir de faux covoiturations il doit aussi entrer son numéro de billet pour le spectacle. Cela permet de vérifier qu'il a bien une place pour l'événement. Malheureusement nous n'avions pas la main sur la billetterie de la salle car elle est gérée par une autre entreprise et nous ne pouvions pas récupérer les numéros de billet pour les spectacles de manière automatique. Donc quand un utilisateur créé un covoiturage, le système doit envoyer un email à la salle de théâtre et quelqu'un doit vérifier manuellement si le billet est valide. Après vérification, la personne peut valider un covoiturage pour qu'il s'affiche.

Pour que les utilisateurs puissent voir les covoiturations qu'ils ont créés et les trajets auxquels ils participent, nous devons créer un système de comptes, avec un page utilisateur qui permet de retrouver tout cela.

Il fallait aussi créer une page qui liste les covoiturages disponibles, avec un moyen de filtrer les covoiturages par date ou spectacle. Pour des raisons de sécurité, nous avons pensé que seules les personnes avec un compte et qui sont connectées peuvent voir la liste des trajets.

Et pour finir nous devions créer un système de réservation. Le choix qui a été fait est le suivant. Quand un utilisateur recherche un covoiturage, il regarde la liste des covoiturages proposés et clique sur celui qui l'intéresse. Il arrive sur une page où l'on voit le détail de ce covoiturage. Il a alors la possibilité d'envoyer un message au conducteur. Pour les messages, le site ne dévoile jamais les données des utilisateurs (email, numéro, etc). Pour cela nous avons suivi l'exemple du site Leboncoin. Lorsqu'un message est envoyé, le destinataire reçoit un email pour le prévenir qu'il a reçu un nouveau message pour un covoiturage. Ensuite il va y répondre sur le site.

## Création des comptes utilisateurs

Après avoir compris tous les objectifs du stage, nous avons commencé par créer le système de compte utilisateur. Wordpress dispose de plusieurs extensions qui créent tout ce système. Nous avons choisi d'utiliser l'extension « Ultimate Member ».

Ultimate Member permet de créer directement des comptes utilisateurs. L'extension crée aussi les pages de connexion, d'inscription, de mot de passe oublié, et la page « mon profil ». Bien sûr ces pages sont simples et nous avons des modifications à apporter sur chacune d'elles. Premièrement, tout était en anglais. Pour la traduction, nous avons utilisé Loco Translate qui est une autre extension qui permet de réaliser plusieurs traductions du site. Comme c'était notre première fois sur WordPress cette mise en place a pris quelques jours mais avec des recherches c'est assez facile à faire.

Le site de billetterie des laboratoires vivants dispose déjà du système de comptes mais nous ne pouvons pas réutiliser ces comptes pour les mêmes raisons que nous n'avons pas le droit de récupérer de façon automatique les numéros de billets. Donc pour éviter les confusions nous avons rajouté dans le code de la page de connexion et d'inscription un message de prévention. Il rappelle aux gens que ce n'est pas le même compte. Nous avons également écrit un message pour la connexion au compte de covoiturage. Voilà les deux formulaires finaux :

### créer un compte

Veuillez vous connecter pour accéder aux covoiturages

Le compte de covoiturage n'est pas le même que le compte de la billetterie.

Prénom

Nom

E-mail

Mot de passe

Confirmer le mot de passe

Créer

Annuler

### connexion

Veuillez vous connecter pour accéder aux covoiturages

Le compte de covoiturage n'est pas le même que le compte de la billetterie.

E-mail

Mot de passe

☐ Se souvenir de moi

Se connecter

Créer un compte


Mot de passe oublié



Ceci n'est que la partie formulaires. Nous les avons intégrés dans les pages que nous avons créées en respectant le design du site. Vous pouvez voir les pages complètes dans les annexes 1 et 2.


La page de réinitialisation de mot de passe est basique. Vous pouvez la voir en annexe 3. Nous avons simplement traduit ultimate member. C'est un simple formulaire qui demande l'email du compte puis envoie un mail à l'utilisateur avec un lien. L'utilisateur crée ensuite un nouveau mot de passe.


La dernière page sur laquelle nous avons travaillé à ce moment-là était la page de profil de l'utilisateur lui permettant de modifier son compte, changer son mot de passe et supprimer son compte. Pour cette page nous avons eu quelques modifications à faire. L'extension proposait automatiquement de faire des posts et d'avoir une photo de profil. Nous avons donc retiré ces possibilités car elles ne nous étaient pas utiles. Cela nous donne la page en annexe 4 :



## compte

**Timeo Danois**  
[Voir profil](#)

 **Compte** ▼


 **Modifier le mot de passe** ▼

**Mot de passe actuel**

**Nouveau mot de passe**

**Confirmez le mot de passe**

**Mettre à jour le mot de passe**

 **Supprimer le compte** ▼

Nous avons ici le nom et prénom de l'utilisateur

On peut cliquer ici et cela déroule un formulaire pour modifier l'email du compte

Vous voyez ici le formulaire déroulé pour modifier le mot de passe

Ici le formulaire pour supprimer le compte

Le lien « voir profil » sous le nom et prénom amène à la page qui liste les covoiturages auxquels l'utilisateur participe. J'explique cela dans la partie « Développement du système de message ».

Nous avons travaillé à deux avec l'aide de notre maître de stage sur cette partie car nous n'avions pas beaucoup de connaissances sur le fonctionnement de wordpress. Mais après avoir mis en place ce système de

compte nous avons déjà bien compris le fonctionnement des plugin wordpress ainsi que des thèmes qu'ils utilisent.

## Création des covoiturages

Après avoir créé les comptes nous avons commencé à faire les covoiturages. Cela s'est fait en plusieurs étapes, comme vous pouvez le voir dans le sommaire. Premièrement nous avons créé les moyens de les stocker, ensuite nous avons fait l'affichage de la liste des trajets. Puis les formulaires pour les créer. Il y a aussi la page avec le détail d'un covoiturage, la modification de celui-ci, et les messages entre les utilisateurs.

Pour commencer nous avons réfléchi à comment les stocker. Alan nous a expliqué que la façon de faire dans WordPress était d'utiliser les plugins ACF et CPT UI qui sont souvent complémentaires et utilisés pour ce type de projet.

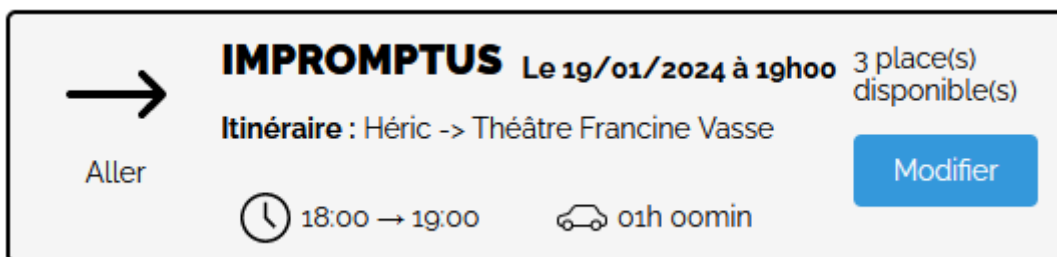
ACF permet de créer des champs comme une table en base de données. Et CPT UI, pour résumer, permet de récupérer ces données facilement pour les afficher directement sur une page.

Donc nous avons créé un type de post « Covoiturage » avec CPT UI, et avec ACF nous avons créé les champs dont nous avons besoin pour stocker les données. Pour commencer nous avons créé 13 champs :

- Événement, qui nous permet de stocker directement un spectacle. Grâce à wordpress nous pouvons directement lier ce champ à un événement des laboratoires vivants.
- Ensuite la date et heure de l'événement qui est exactement ce que le nom laisse penser.
- Offreur du covoiturage, c'est donc ici qu'on stocke l'utilisateur qui propose le covoiturage. Avec wordpress on peut directement enregistrer des comptes dans ACF.
- Nous avons un champs avec trois possibilités entre « Aller »/« Retour »/« Aller – Retour »
- Nous avons ensuite trois champs, l'heure de départ et d'arrivée du trajet. Et si jamais le trajet est « Aller- Retour » il y a « Heure de retour » qui s'active.
- Nous avons la description du trajet qui comme expliqué dans la réunion est simplement une zone de texte. Là où l'utilisateur va parler du chemin qu'il souhaite faire pour aller à la salle de théâtre.
- Comme nous sommes prévoyants, nous avons directement créé un champs « mode de transport » avec covoiturage et copiétonnage.
- Il y a bien sûr le nombre de places disponibles dans la voiture.
- Le numéro de billet qui sert à la vérification du covoiturage.

- Nous avons un Boolean « Complet » qui nous sert à savoir si on affiche le covoiturage dans la liste des trajets et qui permet au conducteur de masquer son covoiturage pour ne plus recevoir de demande s'il n'a plus de place.
- Pour finir nous avons un autre boolean qui permet de savoir si le covoiturage a été validé ou non par le théâtre.

Avec cela nous pouvions créer directement dans wordpress des covoiturages. Nous avons fait ça et nous avons créé un premier affichage d'un covoiturage, avec un design basique. Le model sera changé plus tard dans le stage pour mieux coller à la charte graphique du site.



En faisant cet affichage j'ai compris comment récupérer les données d'un covoiturage. J'ai créé un code qu'on appelle et qui crée juste ce bloc. Ça nous a permis de le réutiliser dans toutes nos autres pages.

Pour la répartition du travail, j'ai travaillé principalement sur les messages et les emails, ainsi que sur la page d'un covoiturage. Tandis que la partie de la liste des trajets et du formulaire d'ajout et celle qui liste les covoiturages d'une personne sur son profil a surtout été traitée par Rémi, l'autre stagiaire.

## Création de la liste des trajets

Cette page doit lister tous les covoiturages futurs disponibles. Il doit y avoir un filtre de recherche et on doit pouvoir cliquer sur les covoiturages pour contacter le créateur du trajet.

Pour sélectionner les covoiturages Alan nous a montré un exemple de requête qu'il utilisait ailleurs. Avec cette méthode on récupère tous les trajets qui sont valides, et non complets, et qui sont dans le futur. Après on réutilise le code qui affiche le bloc d'un covoiturage dans une boucle avec les données de chacun. Ce qui donne ce résultat :

### covoiturage

#### LES TRAJETS :

→

**IMPROMPTUS** Le 19/01/2024 à 19h00 3 place(s) disponible(s)

Itinéraire : Héric -> Théâtre Francine Vasse

18:00 → 19:00 01h 00min

Contactez le

Aller

←

**IMPROMPTUS** Le 19/01/2024 à 19h00 5 place(s) disponible(s)

Itinéraire : Héric -&gt; Théâtre Francine Vasse

17:50 → 18:30 00h 40min

Modifier

Retour

Dans cet exemple nous avons deux covoiturages de test. Le premier est créé par un autre utilisateur et le deuxième par moi. On le remarque car le premier, le site me propose de le contacter alors que le second, il propose de le modifier.

Ce design était temporaire, c'était la première version. Vous pouvez voir en annexe 5 un exemple plus travaillé.

Nous avons mis le lien de cette page dans l'accueil du site. Vous pouvez voir ici « trajets » que nous avons ajouté :

**février**

**trajets**

**une passion dévoilée**  
CLAUDE BRUMACHON ET  
BENJAMIN LAMARCHE

**presqu'un chant,  
un chuchotement...**  
CHAU-CUONG LÊ

**êtres en création  
dustbowl**  
KARIM BOUHEUDJEUR

**vidéothèque**  
**archipel**  
**sorties**  
**laborantines**

**amour h.**  
GR INFINI

**école  
buissonnière**  
**pratiquer**  
**visite virtuelle**

**la chute de l'île  
de la décadanse**  
COMPAGNIE DE L'AUTRUCHE

CATÉGORIE

THÉÂTRE FRANCINE VASSE - LES LABORATOIRES VIVANTS

Nous avons décidé que seuls les utilisateurs connectés avaient accès à cette page. Grâce à trois lignes de code nous redirigeons l'utilisateur vers la page de login s'il n'est pas déjà connecté. L'extension ultimate member propose une fonctionnalité qui le fait automatiquement mais si nous l'activions ça provoquait un bug et le page d'accueil demandait aussi la connexion. Cela venait du lien que nous avons ajouté. Etant donné que le code pour le faire manuellement est simple nous avons choisi cette possibilité.

## Formulaire d'ajout de covoiturage

Nous avons décidé de créer le formulaire d'ajout un peu de la même façon que le code pour afficher un covoiturage. C'est à dire que c'est un code qu'on appelle « part ». C'est un fichier avec du code qu'on appelle via d'autres codes. Cela permet de découper les programmes et de ne pas trop encombrer les codes. C'est plus lisible et en plus on peut les appeler plusieurs fois dans des pages différentes.

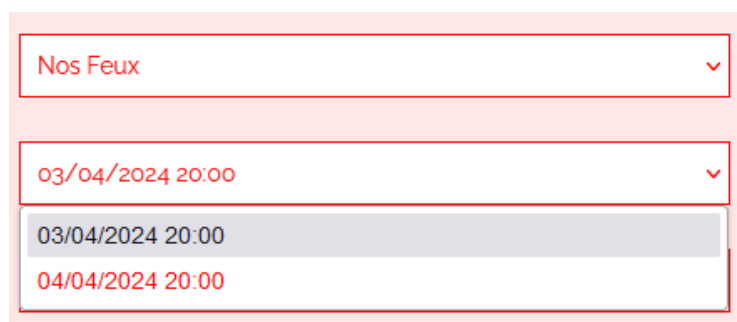
Donc le formulaire d'ajout de covoiturage est un formulaire basique en apparence. Vous pouvez le voir annexe 6. Les difficultés dessus ont été de mettre dans les champs déroulants des événements futurs pour lesquels on peut mettre en place du covoiturage pour le grand public. Car le théâtre propose aussi des spectacles pour les écoles par exemple où, là, on ne peut pas faire de covoiturage comme c'est l'école qui amène les élèves.



The image shows a web form with a dropdown menu. The header of the dropdown is "SELECTIONNER UN EVENEMENT" with a downward arrow. The menu is open, showing a list of events: "Distro", "Restitution Partition", "Restitution Partition", "Itinéraire d'une alchimie", "Dustbowl", "Être en relations", "Nos Feux", "4 954 km de liens", "Soirée Passerelles", "Amorce.s", "Frictions", "Frictions", "Trophée Faltaziñ", and "ÉTRES EN CRÉATION PARTITION". A small "Ver" button is visible on the right side of the menu.

Pour avoir le bon résultat Rémi a dû faire une requête semblable à celle qui récupère les covoiturations mais avec les événements. Avec donc des filtres qui récupèrent juste les événements de la saison en court qui sont pour tout le monde.

L'autre difficulté est que les champs suivants de la date et l'heure de l'événement doivent contenir des dates réelles pour le spectacle sélectionné. Il a donc dû aussi créer une fonction pour ce champs-là.



The image shows two stacked form fields. The top field is a dropdown menu with "Nos Feux" selected and a downward arrow. The bottom field is a date and time selector. It shows "03/04/2024 20:00" with a downward arrow. Below this, a list of options is shown: "03/04/2024 20:00" (highlighted) and "04/04/2024 20:00".

Pour résumer, une fois rempli, le formulaire appelle une fonction qui crée un post ACF avec les données rentrées. Pendant le processus de création nous avons ajouté dans la fonction le fait que le formulaire se désactive et un message de chargement s'affiche (annexe 7), afin de prévenir l'utilisateur que la création est en cours. Ensuite on le redirige vers la page du trajet qu'il vient de créer. Cela envoie aussi un mail au théâtre pour qu'il vérifie manuellement le numéro de billet et valide ou non le trajet. Une fois validé le conducteur reçoit un mail pour le prévenir et son trajet s'affiche dans la liste. Vous verrez plus loin dans le rapport comment fonctionne l'envoi de mail.



## Interface de détails d'un trajet

La page de détail d'un seul trajet sert à beaucoup de choses. C'est sur cette page qu'un utilisateur (ou un administrateur) peut modifier un covoiturage, c'est ici qu'est redirigé le créateur d'un covoiturage. C'est aussi sur cette page qu'il y a les conversations entre utilisateurs. L'interface change donc selon les utilisateurs. Je vais donc expliquer les trois possibilités.

### 1 - Pour une personne qui recherche un covoiturage

L'utilisateur est donc en recherche de covoiturage pour un spectacle. Il s'est connecté à son compte, et a ensuite effectué une recherche dans la liste des trajets. Après avoir cliqué sur le covoiturage qui l'intéresse, il se retrouve sur la page du covoiturage.

CONTACTEZ LE CONDUCTEUR POUR PRÉVOIR UN  
COVOITURAGE   
[Voir profil](#)

 Aller

**IMPROMPTUS** Le 19/01/2024 à 19h00  
Itinéraire : Héric -> Théâtre Francine Vasse  
3 place(s) disponibles  
🕒 18:00 → 19:00 🚗 osh oomin  
Contactez le

Vous pouvez parler sur le site avec les autres utilisateurs sans jamais avoir besoin de dévoiler vos données personnelles.

Envoyer

On Affiche encore une fois le bloc avec toutes les informations du covoiturage. Sur ce screen c'est le premier design. Par la suite il y en aura un nouveau (annexe 8).

Nous avons mis un message de prévention, pour expliquer aux gens qu'ils ne sont pas obligés de divulguer des données personnelles et que le site ne le fera pas pour eux.

Et pour finir l'utilisateur peut écrire et envoyer un message au conducteur. Quand il clique sur envoyer, ça envoie un mail au conducteur pour lui notifier qu'il a un message sur ce covoiturage. De la même façon quand l'offreur du trajet répond le demandeur reçoit un mail qui le prévient qu'une réponse a été donnée.

Une fois qu'une conversation a été commencée l'utilisateur peut retrouver le covoiturage sur sa page profil.

Un bloc avec les messages envoyés entre l'offreur et le demandeur s'affiche maintenant sur la page :

Vous pouvez parler sur le site avec les autres utilisateurs sans jamais avoir besoin de dévoiler vos données personnelles.

The screenshot shows a chat window with a white background. At the top, there is a green message bubble with the text "Bonjour, je suis intéressé pour que vous m'ameniez depuis le cardo." Below it, on the left, is a blue message bubble with the text "Bonjour, oui c'est possible, je peux passer au Cardo à 19h40". To the right of the blue bubble is a green message bubble with the text "Super ! Alors .....". Below the chat bubbles is a text input field with the placeholder text "Ecrivez votre message ici". At the bottom left of the input field is a red button with the text "Envoyer".

Quand il y a beaucoup de messages, le bloc reste d'une hauteur de 500 px maximum et on peut scroller à l'intérieur pour parcourir les messages.

Vous pouvez revoir la page complète en annexe 9.

## 2 – Pour la personne qui propose le trajet

La page est semblable mais il y a quelques différences pour l'offreur du trajet. D'abord lui peut y accéder avant même que l'offre soit validée par un administrateur afin qu'il puisse la modifier s'il y a un problème. Il la retrouve de plusieurs façons. Quand il crée son offre, il est redirigé sur la page simplement, mais il reçoit aussi un mail le prévenant que le trajet est en vérification et lui communiquant le lien du trajet. Il peut aussi la retrouver dans la partie « mes trajets » sur son profil.

(Annexe 10)

Avant la validation de son trajet l'interface a juste le bloc d'informations du covoiturage, avec un message qui prévient que le covoiturage est en cours de validation. Il a ensuite le formulaire pour modifier son covoiturage, et un bouton pour le supprimer.

## VOTRE COVOITURAGE

[Voir profil](#)

Retour	<b>DISTRO</b> Le 12/04/2024 à 20h00 Covoiturage proposé par Stagiaires 2024 Itinéraire : Héric -> Nantes 19:30 → 20:00 00h 30min	3 place(s) disponible(s) <a href="#">Modifier</a>
--------	---	--

Vous pouvez parler sur le site avec les autres utilisateurs sans jamais avoir besoin de dévoiler vos données personnelles.

Rémi r*****@orange.fr
Paul t*****@ecoles-epsi.net
Timeo t*****@gmail.com

## MODIFIER LE COVOITURAGE :

[Supprimer le Covoiturage](#)

Ensuite son covoiturage est validé par le théâtre. Il reçoit donc un mail pour le prévenir. Les utilisateurs peuvent alors lui envoyer des messages. Chaque utilisateur qui lui envoie un message crée une conversation qui va s'afficher sur la page.

Comme vous pouvez le voir, afin que ce soit lisible, nous avons décidé d'afficher le prénom ainsi que l'adresse mail masquée.

Il peut cliquer sur ces conversations pour les lire et y répondre :

Vous pouvez parler sur le site avec les autres utilisateurs sans jamais avoir besoin de dévoiler vos données personnelles.

Rémi r*****@orange.fr
Paul t*****@ecoles-epsi.net
Timeo t*****@gmail.com

Bonjour, je suis intéressé pour que vous m'amenez depuis le cardo.

Bonjour, oui c'est possible, je peux passer au Cardo à 19h40

Super ! Alors ...

Ecrivez votre message ici

Envoyer

Comme précédemment, les blocs sont limités à 500px de hauteur et on peut scroller à l'intérieur. La petite difficulté sur cette partie a été de faire en sorte que quand on ouvre la conversation le scroll se fasse automatiquement tout en bas. Car de base quand elle s'ouvre on est en haut, mais les messages en bas sont les plus récents. Après 3 méthodes qui ne fonctionnaient pas et plusieurs tests j'ai demandé l'aide d'Alan qui a trouvé très vite ce qu'il fallait rajouter dans mon script JavaScript.

Il y a une option que je n'ai pas eu le temps de rajouter sur cette partie malheureusement, c'est le bouton « Signaler », nous en avons parlé sur la fin du stage mais il restait beaucoup de détails à régler et je n'ai pas eu le temps de le développer. Bien sûr il sera ajouté par quelqu'un d'autre que moi. L'idée était de le mettre sur la droite, en face de l'adresse mail. Ce bouton afficherait un formulaire, avec un champs texte pour expliquer la raison du signalement. Le formulaire enverrait un mail au théâtre qui devra regarder si l'utilisateur signalé a fauté et s'il doit avoir une suppression de compte ou autre.

### 3 – Pour un employé des laboratoires vivants

Quand un employé regarde la page, il a accès à toutes les informations. Il peut modifier le covoiturage s'il y a des choses qui ne vont pas, il peut envoyer des messages, et regarder les messages qui posent problème s'il y a eu un signalement. C'est aussi ici qu'il peut valider ou non un covoiturage.

Voilà à quoi ressemble la page :

**CONTACTEZ LE CONDUCTEUR POUR PRÉVOIR UN COVOITURAGE**

[Voir profil](#)

Aller → **NOS FEUX** Le 03/04/2024 à 20h00

proposé par Timeo

5 place(s) disponible(s)

Itinéraire : Blain -> Nantes

🕒 19:35 → 🚶 04h 25min

Modifier

Vous pouvez parler sur le site avec les autres utilisateurs sans jamais avoir besoin de dévoiler vos données personnelles.

Ecrivez votre message ici

Envoyer

Rémi r\*\*\*\*\*@orange.fr

Paul t\*\*\*\*\*@ecoles-epsi.net

Timeo t\*\*\*\*\*@gmail.com

Vous pouvez voir que la page est un mélange des deux versions précédentes. Il peut envoyer un message et ouvrir une conversation avec l'offreur du trajet. Mais il peut aussi modifier le covoiturage.

Mais il peut aussi voir ici les conversations qui se sont créées autour de ce trajet. Cela est utile s'il y a des signalements.

#### MODIFIER LE COVOITURAGE :

**MODIFIER LE COVOITURAGE :**

**Supprimer le Covoiturage**

Complet : Non ☐ Billet : 123456789 Valider : Non ☐

123456789

Blain -> Nantes

5

19:35

--:--

Aller

VALIDER LES MODIFICATIONS

La deuxième nouveauté est le bouton toggle switch ici « valider ». C'est là qu'il peut valider le covoiturage. Grâce à une fonction php appelée en ajax comme la modification du trajet, ce bouton va changer le champs « valide » dans WordPress. Cela va donc permettre au site de l'afficher dans la liste des trajets.

Dans cette partie, j'ai présenté essentiellement le visuel de la page, en évoquant les mails et les messages. Cette page a beaucoup de fonctionnalités, elle a été très intéressante à réaliser car c'est aussi en la développant que la fonction qui envoie les mails a été créée. Dans la partie suivante du rapport de stage je vais donc vous expliquer la suite de cette page, comment a été créé le système de message.

## Développement du système de message et des emails

Quand est arrivé le moment de développer le système de messages, il était obligatoire de développer la fonction qui envoie les emails. C'est pour cela que ces deux parties sont liées. Pour rendre les explications le plus simple possible je vais découper chronologiquement le développement de ces fonctions.

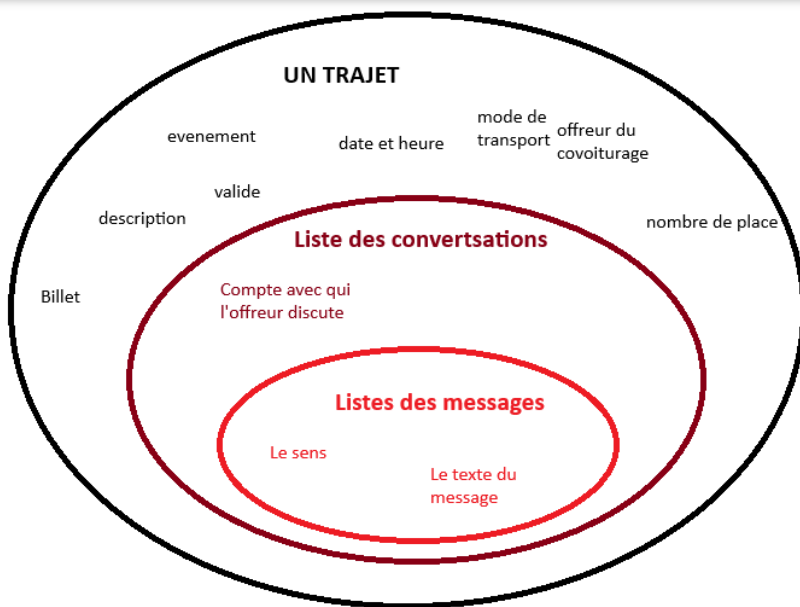
Tout d'abord il a fallu réfléchir à comment stocker les messages dans WordPress. Une fois cette méthode trouvée et mise en place, il fallait développer les fonctions qui allaient enregistrer les messages quand un utilisateur envoie un message à un autre. Une fois cela fait, et que les messages s'enregistraient et s'affichaient, il fallait prévenir le destinataire. Et nous avons décidé avec le théâtre que c'était avec un email. La création de la fonction qui envoie les mails a été réalisée. Pour finir quand tout le système de messages était fonctionnel, nous avons réutilisé la fonction d'email pour tous les autres endroits où cela était utile.

### 1 – Le stockage des messages

Pour stocker les messages nous sommes retournés dans le groupe de champs ACF que nous avons créé au début du stage pour stocker les covoiturages. Nous avons rajouté un champ, qui contient la liste des messages. Dans ce champ nous stockons pour chaque covoiturage toutes les conversations qu'il y a pu avoir pour celui-ci. C'est possible grâce à un type de champ qui s'appelle « répéteur ». C'est en fait un groupe de champs dans notre groupe de champ.

Ensuite dans ce champ de type répéteur nous stockons deux champs. Le premier est le compte avec lequel la conversation est faite. Ce champ est du même type que celui qui stocke le compte de l'offreur du trajet. Ainsi on sait que la conversation est entre l'offreur du covoiturage et le compte enregistré dans le répéteur. Le deuxième champ est un autre répéteur qui contient les messages de la conversation. Il contient lui aussi deux champs, le premier le sens de la conversation, qui est « Demandeur -> Offreur » ou « Offreur -> Demandeur ». Le deuxième est le texte du message.

Voilà un schéma du groupe de champs dans WordPress :



Nous avons d'abord le groupe de champs qui enregistre un trajet.

Dedans il y a des champs simples (l'événement, mode de transport, etc)

Et il y a le champ répéteur qui stocke les conversations.

Un trajet peut donc avoir une multitude de conversations grâce au champ répéteur. Et dans chacune de ces conversations il y a un autre champ répéteur qui permet d'avoir une multitude de messages pour chaque conversation.

## 2 – Affichage et enregistrement des messages

Pour l'affichage des conversations comme vous avez vu précédemment c'est plutôt simple. Sur la page d'un covoiturage nous avons déjà récupéré toutes les données du trajet. Les champs de type répéteur sont traités comme des tableaux. Donc pour afficher toutes les conversations on se sert d'une boucle « foreach » pour faire chacune des conversations, et on refait dedans une autre boucle « foreach » pour parcourir chacun des messages. Le sens des messages nous sert à savoir si les messages vont à gauche ou à droite. Le reste de la mise en page c'est du CSS et de HTML, avec un peu de JavaScript pour que la conversation s'ouvre et scrolle.

Pour que les messages s'envoient et s'enregistrent nous avons créé deux fonctions php. Une pour le sens « Offreur -> Demandeur » et une pour « Demandeur -> Offreur ». Les fonctions sont appelées en ajax grâce à un formulaire qui contient un champs texte et l'id du post de covoiturage. La différence est que quand c'est un conducteur qui répond à un message on récupère aussi dans le formulaire le numéro de la conversation à laquelle il répond. Exemple de la requête qui appelle les fonctions php en annexe 11.

Ensuite dans le fichier fonctions.php nous avons créé les deux fonctions qui se ressemblent (annexe 12).

Pour le sens Offreur -> Demandeur :

On récupère les informations du covoiturage, on vérifie que l'utilisateur connecté est bien l'offreur du covoiturage. Ensuite on met à jour en ajoutant le message avec le sens offreur – demandeur dans le répéteur de la conversation qui a l'id que nous avons envoyé dans le formulaire. Pour finir on appelle la fonction qui envoie un email au destinataire du message.

Pour le sens Demandeur -> Offreur :

On récupère les informations du covoiturage et de l'utilisateur connecté. On vérifie que ce n'est pas le conducteur qui s'envoie un message à lui-même. Ensuite on parcourt les conversations pour voir s'il y a déjà une conversation entre l'utilisateur connecté et le conducteur sur ce covoiturage. Si c'est le cas alors on récupère l'id de la conversation et on ajoute le nouveau message. Sinon on crée une conversation avec l'utilisateur et son message. Pour finir on envoie un email au conducteur pour le prévenir qu'il y a un nouveau message.

Une fois que les fonctions se sont exécutées la page se recharge et les conversations sont donc mises à jour avec les nouveaux messages.

### 3 – Création des emails et de la fonction pour les envoyer.

Pour le système d'email nous avons créé un groupe de champs ACF sur wordpress intitulé « emails transactionnels », et nous avons lié ce groupe de champs à la page option de wordpress. Dans ce groupe nous avons créé 1 champs par email dont nous avons besoin. Ces champs contiennent deux champs texte, un pour le sujet et un pour le texte du mail. Annexe 13.

Maintenant grâce à ce groupe de champs on peut écrire et modifier les emails dans la page option de wordpress. Car cela a ajouté une partie « Emails transactionnels » avec les emails dedans, Annexe 14. Ainsi les laboratoires vivants peuvent modifier les emails dans wordpress. Et nous avons décidé que certaines variables utiles que nous avons listées pouvaient être appelées dans les emails en écrivant « [nom\_de\_la\_variable] ». Par exemple le premier mail qu'on voit dans l'annexe 14 est l'email envoyé au



théâtre pour les prévenir qu'il y a une nouvelle offre de covoiturage. Il contient toutes les informations de cette nouvelle offre et donc plusieurs variables entre crochets (nom de l'offreur, dates, mode de transport, etc).

Nous avons créé une fonction php dans « fonctions.php » qui remplace les variables entre crochets par le texte demandé et qui envoie l'email (annexe 15). On appelle la fonction avec quatre arguments, comme ceci :

```
$email_infos = array(
    'message' => $_POST['message'],
    'offre_covoiturage' => $offre_covoiturage,
    'lien_offre_covoiturage' => get_permalink($post_id),
);

// Récupérer l'auteur du post d'offre de covoiturage
$destinataire = get_field( selector: 'offreur_covoiturage', $post_id);
$to = $destinataire['user_email'];

//fonction pour envoyer un mail
envoyerEmail( valeur: 'nouveau_message_covoiturage', $destinataire, $to, $email_infos);
```

Le premier est le nom de l'email qu'on envoie (celui qui est dans emails transactionnels)

Le deuxième argument est le compte du destinataire, et le troisième est directement son email.

Le dernier argument est un tableau dans lequel on met les informations utiles à l'email pour remplir les potentiels variables qu'il y a dedans. Dans notre exemple, il y a le message envoyé, l'offre de covoiturage et le lien de cette offre.

Dans la fonction « envoyerEmail » (annexe 15) on récupère le sujet et le texte de l'email demandé (ligne 167/168). Et pour les deux on remplace les « [variables] » par les informations qui vont dedans. Pour finir avec wp\_mail qui est une fonction php on envoie l'email final au destinataire.

## Création de la page de nos covoiturages

Quand un utilisateur poste un covoiturage ou discute avec un conducteur pour participer à un trajet il peut retrouver ces trajets en passant par son profil. Cette page permet de trouver ses trajets à venir et aussi ses trajets passés.

### mon compte

[Modifier mon profil](#)  
**mes trajets à venir**

Aller  
→

**4 954 km de liens** Le 21/03/2024 à 00h00  
proposé par Stagiaires 2024  
**Itinéraire :** aaaaaaa  
🕒 22:00 → 23:40 🚶 01h 40min

3 place(s) disponible(s)  
[Modifier](#)

Retour  
←

**distro** Le 12/04/2024 à 20h00  
Covoiturage proposé par Stagiaires 2024  
**Itinéraire :** Héric -> Nantes  
🕒 19:30 → 20:00 🚗 00h 30min

3 place(s) disponible(s)  
[Modifier](#)

Aller -  
Retour  
→  
←

**frictions** Le 13/04/2024 à 20h30  
Covoiturage proposé par Stagiaires 2024  
**Itinéraire :** Je pars de La Chapelle sur Erdre pour aller jusqu'à la salle. Je peux passer par Orvault (Le Cardo), comme par la Beaujoire. Je peux faire de petits détours dans la ville pour aller chercher quelqu'un comme le redéposer après le spectacle. Je resterai sûrement boire un verre à la fin du spectacle, c'est pourquoi l'heure du retour est plus tard que la fin réelle du spectacle.  
🕒 19:50 → 20:30 🚗 00h 40min

5 place(s) disponible(s)  
[Modifier](#)

**mes trajets passés**  
Vous n'avez pas encore fait de trajets.

Pour afficher les trajets il y a donc deux requêtes différentes, une pour les trajets futurs, et une pour les trajets passés.

Les requêtes sont simples, on récupère les trajets où le conducteur est égal à l'utilisateur ou les trajets qui ont l'utilisateur dans le sous-champ compte du répéteur de la liste des messages. Il y a aussi une relation « AND » pour les deux qui est différente selon la requête. Elle concerne la date des événements des trajets. Pour les trajets futurs on sélectionne ceux où la date de l'évènement est supérieur ou égal à la date d'aujourd'hui et on fait donc l'inverse pour ceux passés en prenant les dates inférieures ou égales.

## Mise en place des messages administrables

Nous avons mis des messages de préventions ou d'explications à plusieurs endroits dans les différentes pages. Par exemple sur la page de connexion (annexe 1) nous avons deux messages, un qui demande de se connecter pour voir les trajets, et un autre qui rappelle que le compte pour la billetterie et celui des covoiturages sont différents.

Sur la fin du stage on nous a demandé de faire en sorte que ces textes soient administrables par le théâtre. Ainsi ils peuvent les changer si les textes ne leur conviennent plus ou même rajouter des phrases s'ils doivent prévenir les utilisateurs d'une nouveauté ou d'un problème.

Pour ce faire on a repris la méthode des emails transactionnels en créant un groupe de champs ACF lié à la page option. Dedans on a donc mis de simples champs qui sont des éditeurs de texte. Après avoir listé les textes administrables nous en avons 10 (annexe 16). Seul le message d'alerte de suppression de covoiturage est d'un type différent. C'est le message d'alerte du pop-up qui s'affiche pour demander confirmation quand on veut supprimer son covoiturage. Il est appelé avec du javascript et avec le type éditeur WYSIWYG le message ne s'affichait pas. Cela est dû au fait que l'éditeur WYSIWYG renvoie du code en html ce qui permet de mettre du texte en gras, italique, changer la taille etc. Mais ça ne fonctionnait pas avec le message de confirmation, donc le type texte renvoie juste les caractères entrés.

Ensuite il a suffi de modifier les endroits où nous avons écrit les textes en dur en html par l'appel du message concerné.

Par exemple dans la page connexion avant :

```
20 <div class="um <?php echo esc_attr( $this->get_class( $mode ) ); ?> um-<?php echo esc_attr( $form_id ); ?>">
21 <p>Veuillez vous connecter pour accéder aux covoiturages</p>
22 <p>Le compte de covoiturage n'est pas le même que le compte de la billetterie.</p>
23 <div class="um-form">
```

Après :

```
20 <div class="um <?php echo esc_attr( $this->get_class( $mode ) ); ?> um-<?php echo esc_attr( $form_id ); ?>">
21 <?php echo get_field( selector: 'connexion', post_id: 'option' ); ?>
22 <div class="um-form">
```

## Conclusion

J'ai beaucoup appris pendant ce stage avec Alan ferrennière. J'ai principalement découvert Wordpress, c'est un système très pratique que je réutiliserai peut-être dans certains de mes futurs projets car il permet de développer rapidement et facilement grâce aux nombreuses extensions. J'ai aussi travaillé pour la première fois avec Azur Devops qui est un environnement de travail commun. Il permet de faire des repos Git et des board de travail. Le repo Git nous a permis de travailler ensemble et de mettre notre code en commun simplement en travaillant sur différentes branches. Azur Devops permet de créer des branches de code qui se séparent du reste du code. Ainsi chacun travaille de son côté sans s'impacter. C'est un outil très utile utilisé en entreprise et que nous commençons d'ailleurs à utiliser en cours. Ce stage m'a aussi permis de me perfectionner en php, que j'utilise souvent et d'améliorer mon JavaScript que j'avais rarement utilisé auparavant.

Nous avons beaucoup travaillé au cours des 5 semaines de stages, nous avons quasiment fini tout le back mais nous n'avons pas eu le temps de faire le système de signalement. Durant la fin de ma dernière semaine nous avons rencontré le graphiste qui travaille avec les laboratoires vivants et Alan Ferronnière pour qu'il nous donne ses directives sur le design de nos pages. Remi avait lui 8 semaines de stage donc moi je n'ai pas eu l'occasion de travailler sur le design final étant donné que mon stage se terminait. C'est donc lui qui a retravaillé le CSS des formulaires, des blocs de covoiturages, et des pages en générale.

Je remercie encore toutes les personnes qui ont contribué au bon déroulement de ce stage, pour m'avoir épaulé et aidé à approfondir et à mettre en pratique mes connaissances.

## Annexes

### Annexe 1 : Page de connexion

**connexion**

Veuillez vous connecter pour accéder aux covoiturages

Le compte de covoiturage n'est pas le même que le compte de la billetterie.

E-mail

Mot de passe

☐ Se souvenir de moi

[Se connecter](#) [Créer un compte](#)

[Mot de passe oublié](#)

THÉÂTRE FRANCINE VASSE - LES LABORATOIRES VIVANTS

### Annexe 2 : Page d'inscription

**créer un compte**

Veuillez vous connecter pour accéder aux covoiturages

Le compte de covoiturage n'est pas le même que le compte de la billetterie.

Prénom

Nom

E-mail

Mot de passe

Confirmer le mot de passe

[Se connecter](#) [Créer un compte](#)

[Mot de passe oublié](#)

THÉÂTRE FRANCINE VASSE - LES LABORATOIRES VIVANTS

### Annexe 3 Page de réinitialisation de mot de passe

**mot de passe oublié**

Pour réinitialiser votre mot de passe, veuillez saisir votre adresse e-mail ou votre nom d'utilisateur ci-dessous.

Entrez votre email

Réinitialiser le mot de passe

THÉÂTRE FRANCINE VASSE - LES LABORATOIRES VIVANTS

### Annexe 4 : Page profil

**compte**

Timeo Danois  
[Voir profil](#)

Compte

Identifiant

timeo.danois

Adresse e-mail

timeo.dns@gmail.com

Mot de passe


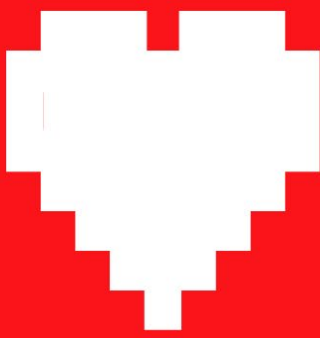
Mettre à jour le compte

Modifier le mot de passe

Supprimer le compte

THÉÂTRE FRANCINE VASSE - LES LABORATOIRES VIVANTS

## Annexe 5 : Liste des covoiturages disponibles



### trajets

LES TRAJETS :

Aller  
→

**AMORCE.S** Le 16/05/2024 à 19h00  
Rémi r...@orange.fr  
Itinéraire : Trentemoult -> Pirmil -> Commerce -> Francine Vasse  
18:00 → 18:45 00h 45min  
8 place(s) disponible(s)  
Contactez le

Retour  
←

**DISTRO** Le 12/04/2024 à 20h00  
Stagiaires 2024 a...@gmail.com  
Itinéraire : Héric -> Nantes  
19:30 → 20:00 00h 30min  
3 place(s) disponible(s)  
Contactez le

Aller  
→

**RUY BLAS** Le 05/05/2024 à 15h30  
Rémi r...@orange.fr  
Itinéraire : St Herblain (Atlantis) -> Théâtre Francine Vasse  
3 place(s) disponible(s)  
Contactez le

THÉÂTRE FRANCINE VASSE - LES LABORATOIRES VIVANTS

THÉÂTRE FRANCINE VASSE

BILLETTERIE

INFORMATIONS PRATIQUES

## Annexe 6 : Formulaire d'ajout de covoiturage

### PROPOSER UN TRAJET



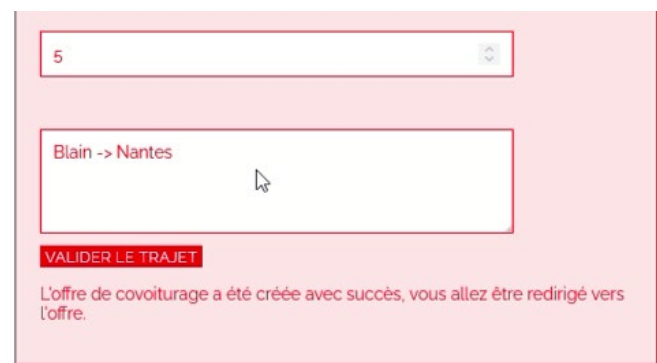
Form for proposing a ride-sharing trip. The form is titled "PROPOSER UN TRAJET" and contains several input fields and a submit button.

- NUMÉRO DE BILLET
- Destination: Aller
- SELECTIONNER UN MODE DE TRANSPORT
- Nos Feux
- Date and Time: 03/04/2024 20:00
- Time zone: -- : --
- NOMBRE DE PLACES
- DÉCRIVEZ LES ÉTAPES DE VOTRE TRAJET
- VALIDER LE TRAJET

## Annexe 7 : Messages pendant la création d'un trajet



Form during trip creation. The form shows the number of seats (5) and the destination (Blain -> Nantes). The submit button is labeled "VALIDER LE TRAJET". Below the button, a message states: "Votre covoiturage est en cours de création..".



Confirmation message during trip creation. The form shows the number of seats (5) and the destination (Blain -> Nantes). The submit button is labeled "VALIDER LE TRAJET". Below the button, a message states: "L'offre de covoiturage a été créée avec succès, vous allez être redirigé vers l'offre."



## Annexe 8 : Interface d'un covoiturage aux couleurs du site



**CONTACTEZ LE CONDUCTEUR POUR PRÉVOIR  
UN COVOITURAGE**

[Voir profil](#)

Aller  
→

**RUY BLAS** Le 05/05/2024 à 15h30  
Covoiturage proposé par Rémi  
**Itinéraire :** St Herblain (Atlantis) -> Théâtre  
Francine Vasse  
🕒 14:45 → 15:10 🚗 ooh 25min

3 place(s)  
disponible(s)  
  
Contactez le

Vous pouvez parler sur le site avec les autres utilisateurs sans jamais avoir besoin de dévoiler vos données personnelles.

Ecrivez votre message ici

Envoyer

## Annexe 9 : Interface des messages du côté d'un demandeur de covoiturage

### CONTACTEZ LE CONDUCTEUR POUR PRÉVOIR UN COVOITURAGE

[Voir profil](#)

Retour ←	<b>DISTRO</b> Le 12/04/2024 à 20h00 Covoiturage proposé par Stagiaires 2024 <b>Itinéraire :</b> Héric -> Nantes 🕒 19:30 → 20:00 🚗 00h 30min	3 place(s) disponible(s) 💬 Contactez le
-------------	--	---

Vous pouvez parler sur le site avec les autres utilisateurs sans jamais avoir besoin de dévoiler vos données personnelles.

Bonjour, je suis intéressé pour que vous m'améniez depuis le cardo.

Bonjour, oui c'est possible, je peux passer au Cardo à 19h40

Super ! Alors .....

Envoyer

## Annexe 10 : Interface d'un covoiturage vu par l'offreur de celui-ci avant sa validation par un administrateur

### VOTRE COVOITURAGE

Votre covoiturage est en attente de validation, les utilisateurs du site ne peuvent pas encore le voir. Vous recevrez un mail quand il sera validé.

[Voir profil](#)

Aller →	<b>NOS FEUX</b> Le 03/04/2024 à 20h00 proposé par Timeo <b>Itinéraire :</b> Blain -> Nantes 🕒 19:35 → 🚶 04h 25min	5 place(s) disponible(s) <b>Modifier</b>
------------	--	---

Vous pouvez parler sur le site avec les autres utilisateurs sans jamais avoir besoin de dévoiler vos données personnelles.

Aucun message pour le moment.

### MODIFIER LE COVOITURAGE :

### MODIFIER LE COVOITURAGE :

[Supprimer le Covoiturage](#)

Complet : Non ☐

123456789
Blain -> Nantes
5
19:35
--:--
Aller
VALIDER LES MODIFICATIONS

## Annexe 11 :

```
// Message dans le sens Offreur -> Demandeur
jQuery(document).ready(function($) {
    $('#covoiturage-msg-rep-form').on('submit', function(e) {
        e.preventDefault();
        var postId = <?php echo get_the_ID() ?>;
        let ajaxurl = "<?php echo admin_url('admin-ajax.php'); ?>";
        var formData = $(this).serializeArray();
        formData.push({ name: 'action', value: 'message_rep_covoiturage' });
        formData.push({ name: 'post_id', value: postId }); // Ajout de post_id ici

        $.ajax({
            type: 'POST',
            url: ajaxurl,
            data: formData,
            success: function(response) {
                console.log('Message envoyé avec succès.');
                window.location.reload();
            },
            error: function(error) {
                console.error('Erreur lors de la mise à jour des champs de covoiturage.', error);
            }
        });
    });
});
```

## Annexe 12 : Fonctions php pour les messages entre utilisateurs

```
// Fonction pour envoyer un message (Offreur-Demandeur)
function message_rep_covoiturage() {
    if (isset($_POST['action']) && $_POST['action'] == 'message_rep_covoiturage') {

        if ($_POST['message'] != '') {
            // Obtenez les données du formulaire
            $post_id = $_POST['post_id'];
            //on recupere les infos de l'offre de covoiturage
            $offre_covoiturage = get_fields($post_id);

            // Récupérer les messages existants
            $liste_messages = $offre_covoiturage['liste_messages'];

            // Récupérer l'auteur du post d'offre de covoiturage
            $offreur_covoiturage_id = $offre_covoiturage['offreur_covoiturage']['ID'];

            // Vérifier si l'utilisateur actuel est l'offreur
            $current_user = wp_get_current_user();
            $current_user_id = $current_user->ID;

            if (($current_user_id == $offreur_covoiturage_id || in_array('administrator', $current_user->roles) && $_POST['id_conversation'] >

                // On enregistre les messages dans la liste des messages
                $liste_messages[$_POST['id_conversation']]['messages'][] = array(
                    'sens' => 'Offreur-Demandeur',
                    'message' => $_POST['message'],
                );

            // Mettre à jour le champ ACF "liste_messages" avec les nouvelles données
```

```

        // Mettre à jour le champ ACF "liste_messages" avec les nouvelles données
        update_field( 'selector: liste_messages', $liste_messages, $post_id);

        // Email
        $email_infos = array(
            'message' => $_POST['message'],
            'offre_covoiturage' => $offre_covoiturage,
            'lien_offre_covoiturage' => get_permalink($post_id),
        );

        // Récupérer le destinataire
        $destinataire = $liste_messages[$_POST['id_conversation']]['compte'];
        $to = $destinataire['user_email'];

        //fonction pour envoyer un mail
        envoyerEmail( valeur: 'nouveau_message_covoiturage_2', $destinataire, $to, $email_infos);

    } else {
        wp_send_json_error( data: 'Vous ne pouvez pas envoyer de message.');
    }
} else {
    wp_send_json_error( data: 'Erreur message vide.');
}
}

} else {
    wp_send_json_error( data: 'Erreur lors de la mise à jour des champs de covoiturage.');
}
}

add_action('wp_ajax_message_rep_covoiturage', 'message_rep_covoiturage');
add_action('wp_ajax_nopriv_message_rep_covoiturage', 'message_rep_covoiturage');

```

```

// Fonction pour envoyer un message (Demandeur->Offreur)
function message_covoiturage() {
    if (isset($_POST['action']) && $_POST['action'] == 'message_covoiturage') {

        if ($_POST['message'] != '') {
            // Obtenez les données du formulaire
            $post_id = $_POST['post_id'];
            //on recupere les infos de l'offre de covoiturage
            $offre_covoiturage = get_fields($post_id);

            // Récupérer les messages existants
            $liste_messages = $offre_covoiturage['liste_messages'];

            // Récupérer l'auteur du post d'offre de covoiturage
            $offre_covoiturage_id = $offre_covoiturage['offre_covoiturage']['ID'];

            // Vérifier si l'utilisateur actuel n'est pas l'offreur
            $current_user = wp_get_current_user();
            $current_user_id = $current_user->ID;

            if ($current_user_id != $offre_covoiturage_id) {

                // Vérifier si l'utilisateur a déjà envoyé un message
                $existing_message_index = false;

                foreach ($liste_messages as $index => $message) {
                    if ($message['compte']['ID'] == $current_user_id) {
                        $existing_message_index = $index;
                        break;
                    }
                }
            }
        }
    }
}

```

```

        break;
    }
}

// Si l'utilisateur a déjà envoyé un message, ajouter le nouveau message à son compte
if ($existing_message_index !== false) {
    $liste_messages[$existing_message_index]['messages'][] = array(
        'sens' => 'Demandeur-Offreur',
        'message' => $_POST['message'],
    );
} else {
    // Si l'utilisateur n'a pas encore envoyé de message, créer un nouveau répéteur
    $nouveau_message = array(
        'compte' => $current_user,
        'id_compte' => $current_user_id,
        'messages' => array(
            array(
                'sens' => 'Demandeur-Offreur',
                'message' => $_POST['message'],
            ),
        ),
    );

    $liste_messages[] = $nouveau_message;
}

// Mettre à jour le champ ACF "liste_messages" avec les nouvelles données
update_field( 'selector' => 'liste_messages', $liste_messages, $post_id );
} else {
    wp_send_json_error( data: 'Vous ne pouvez pas envoyer de message en tant qu\'offreur de covoiturage.' );
}

```

```

$email_infos = array(
    'message' => $_POST['message'],
    'offre_covoiturage' => $offre_covoiturage,
    'lien_offre_covoiturage' => get_permalink($post_id),
);

// Récupérer l'auteur du post d'offre de covoiturage
$destinataire = get_field( 'selector' => 'offreur_covoiturage', $post_id );
$to = $destinataire['user_email'];

//fonction pour envoyer un mail
envoyerEmail( valeur: 'nouveau_message_covoiturage', $destinataire, $to, $email_infos );

} else {
    wp_send_json_error( data: 'Erreur message vide.' );
}

} else {
    wp_send_json_error( data: 'Erreur lors de la mise à jour des champs de covoiturage.' );
}
}

add_action( 'wp_ajax_message_covoiturage', 'message_covoiturage' );
add_action( 'wp_ajax_nopriv_message_covoiturage', 'message_covoiturage' );

```

## Annexe 13 : Groupes de champs ACF « Emails Transactionnels »

Modifier le groupe de champs

Emails transactionnels

+ Ajouter un champ

[Oui](#)  
[Not Really](#)

Champs

N°	Libellé	Nom	Type
1	<a href="#">Email demande validation Admin</a>	email_demande_validation_admin	Groupe
2	<a href="#">Nouveau message covoiturage</a>	nouveau_message_covoiturage	Groupe
3	<a href="#">Nouveau message covoiturage 2</a>	nouveau_message_covoiturage_2	Groupe
4	<a href="#">offre de covoiturage validée (utilisateur)</a>	offre_de_covoiturage_validee_utilisateur	Groupe
5	<a href="#">offre de covoiturage supprimée (offreur)</a>	offre_de_covoiturage_supprimee_offreur	Groupe
6	<a href="#">offre de covoiturage supprimée (demandeur)</a>	offre_de_covoiturage_supprimee_demandeur	Groupe

2 Nouveau message covoiturage

nouveau\_message\_covoiturage

Groupe

Général Validation Présentation Logique conditionnelle

Type de champ

Groupe

Parcourir les champs

Libellé du champ

Nouveau message covoiturage

Ceci est le nom qui apparaîtra sur la page de modification

Nom du champ

nouveau\_message\_covoiturage

Un seul mot. Aucun espace. Les tirets bas et les tirets sont autorisés.

Typographie

Ne pas filtrer

Sélectionner le filtre wp-Typography à appliquer

Sous-champs

Champs

+ Ajouter un champ

N°	Libellé	Nom	Type
1	<a href="#">Sujet</a>	sujet	Texte
2	<a href="#">Contenu</a>	contenu	Éditeur WYSIWYG

+ Ajouter un champ



## Annexe 14 : Aperçu de la partie emails transactionnels dans options

FlyerCatalogueDiapoMédiasEnseignesPagesCommentairesPopup Maker2ListesContenusVidéosPièces 360°Visites 360°Groupes de contenuCovoituragesUltimate MemberWPFormsApparenceExtensions12ComptesOutilsRéglagesACFOptionsEasy WP SMTPCookiesYoast SEOCPT UILoco TranslateRéduire le menu

Emails transactionnelsEmail demande validation AdminSujetUne nouvelle offre de covoiturage est en attente de validationContenuAjouter un médiaVisuelTexteParagrapheB I Paragraph Une nouvelle offre de covoiturage a été publié par [offreur\_covoiturage] pour l'événement "[evenement]" du [date\_heure\_evenement]. Cliquer sûr [lien\_offre\_covoiturage] pour voir la page de l'offre, la valider ou la modifier.Voilà les détails de l'offre :Numéro de billet du conducteur : [billet]Événement : [evenement] le [date\_heure\_evenement]Trajet : [type\_de\_trajet]Heure de départ : [heure\_de\_depart] Heure d'arrivée : [heure\_darrivee]Si aller/retour heure de retour : [heure\_de\_retour]Nombre de places : [nombre\_de\_place]Mode de transport :[mode\_de\_transport]Description : [description]

Nouveau message covoiturageSujetUn utilisateur veut participer à votre covoiturageContenuAjouter un médiaVisuelTexteParagrapheB I [user\_name], vous avez un nouveau message pour le covoiturage que vous avez proposé ([evenement]). Voilà le message : [message]Pour répondre aux messages, vous pouvez passer directement par ce lien [lien\_offre\_covoiturage] ou en passant par votre compte.

## Annexe 15 : Fonction pour envoyer les emails

Par souci de place il n'y a pas toutes les modifications de variables de l'email.

```
162 function envoyerEmail($valeur, $destinataire, $to, $infos){
163     if ($valeur != '' && $to != '' && $destinataire != '') {
164
165         $headers = array('Content-Type: text/html; charset=UTF-8');
166
167         $message = get_field($valeur, post_id: 'option')['contenu'];
168         $subject = get_field($valeur, post_id: 'option')['sujet'];
169
170         $message = str_replace( search: '[titre]', $subject, $message);
171
172         //[user_name]
173         $message = str_replace( search: '[user_name]', $destinataire['display_name'], $message);
174         $subject = str_replace( search: '[user_name]', $destinataire['display_name'], $subject);
175
176         //[utilisateur_nom]
177         $message = str_replace( search: '[utilisateur_nom]', $destinataire['user_lastname'], $message);
178         $subject = str_replace( search: '[utilisateur_nom]', $destinataire['user_lastname'], $subject);
179
180         //[utilisateur_prenom]
181         $message = str_replace( search: '[utilisateur_prenom]', $destinataire['user_firstname'], $message);
182         $subject = str_replace( search: '[utilisateur_prenom]', $destinataire['user_firstname'], $subject);
183
184         //[user_mail]
185         $message = str_replace( search: '[user_mail]', $destinataire['user_email'], $message);
186         $subject = str_replace( search: '[user_mail]', $destinataire['user_email'], $subject);
187
188         if (isset($infos['lien_offre_covoiturage'])) {...}
189
190         //[evenement]
191         if (isset($infos['offre_covoiturage'])) {...}
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```



## Annexe 16 : Groupe de champs ACF des messages administrables et affichage dans options pour modifier les messages

Modifier le groupe de champs Messages administrables covoiturage

[+ Ajouter un champ](#)

Enregistrer les modifications

Champs			
N°	Libellé	Nom	Type
1	création de compte	creation_de_compte	Éditeur WYSIWYG
2	connexion	connexion	Éditeur WYSIWYG
3	Titre de la page d'un covoiturage (offreur)	titre_de_la_page_dun_covoiturage_offreur	Éditeur WYSIWYG
4	Titre de la page d'un covoiturage (demandeur)	titre_de_la_page_dun_covoiturage_demandeur	Éditeur WYSIWYG
5	Message d'attente de validation	message_dattente_de_validation	Éditeur WYSIWYG
6	Au-dessus des messages d'un covoiturage	au-dessus_des_messages_dun_covoiturage	Éditeur WYSIWYG
7	Message d'alerte de suppression de covoiturage	message_dalerte_de_suppression_de_covoiturage	Texte
8	Message d'attente de création de covoiturage 1	message_dattente_de_creation_de_covoiturage_1	Éditeur WYSIWYG
9	Message d'attente de création de covoiturage 2	message_dattente_de_creation_de_covoiturage_2	Éditeur WYSIWYG
10	Message d'attente de création de covoiturage 3	message_dattente_de_creation_de_covoiturage_3	Éditeur WYSIWYG

[illegible]