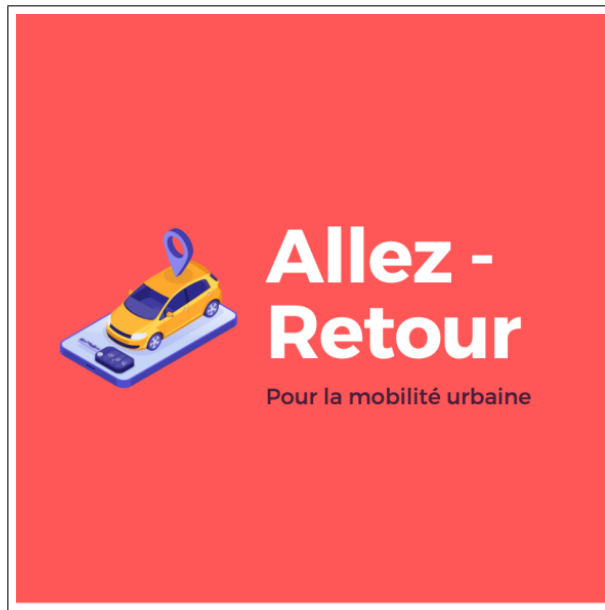


Projet Web

Micciche Luca - Godard Loan - Tharreau Victor



1 Introduction

Dans le cadre des mobilités rurales, nous avons décidé de développer Aller-retour, un site dédié à l'aide au déplacement en zone rurale. L'objectif de cette plateforme est de permettre à des personnes qui ne dispose pas du permis, ou qui n'ont pas accès aux transports en commun, de profiter des trajets réalisés par les villageois. Ainsi ils seraient aptes à se déplacer vers les lieux qu'ils souhaitent, grâce à l'entraide. Le projet est voué par ailleurs à permettre de renouer les liens en zone rurale, liens qui se sont perdus au fur et à mesure de l'urbanisation qu'a connue notre époque et de la perte d'intérêt pour les producteurs locaux lié aux grandes surfaces.

A lire : En ce qui concerne le lancement voici comment faire :

- dans un premier temps, il faut créer la bdd avec *make db.init*
- ensuite on lance le serveur avec *make*
- ensuite pour accéder au site, il faut se rendre à l'adresse suivante : **http://localhost:8080/projet-web/index.php**. En effet, nous avons développé sur WAMP / MAMP et nous n'avons pas de chemin `./public/...` C'est pourquoi notre structure n'est pas très conventionnelle, mais nous retenons cette structure pour nos futurs projets.
- si le site web affiche une bdd non fonctionnelle, il faudra peut-être modifier le nom utilisateur et le mdp postgres dans le fichier `public/projet-web/includes/dbh.inc.php`
- Voici les codes pour se connecter au site avec un compte administrateur / utilisateur :
superadmin email : admin@gmail.com MDP : Micciche98
utilisateur allezretourevry@gmail.com MDP : Micciche98

2 L'organisation du projet

2.1 Organisation

Pour la réalisation de ce site, nous avons procédé dans un premier temps à la réalisation d'un cahier des charges. En effet, bien définir son besoin est primordial pour pouvoir atteindre un objectif. Pour cela nous avons rédigé un document qui détaille les attentes vis-à-vis du projet. Suite à cela, nous sommes passé à la conception de la base de données, sans laquelle on ne peut pas faire grand-chose. Nous avons essayé d'imaginer tous les cas possibles afin de "mâcher" le travail pour plus tard. Néanmoins, nous avons tout de même dû procéder à quelques modifications tout au long du projet. Une fois ces deux parties réalisées, la base était là. Il ne restait plus qu'à commencer à coder le site. Pour répartir les tâches, nous avons décidé d'utiliser un Trello (plateforme web qui permet à des utilisateurs de planifier des tâches à faire pour chaque personne). Une fois réalisé nous avons pu débiter la conception. Malgré le confinement, la communication était bonne, nous communiquions principalement via discord. Détaillons maintenant comment le site est structuré.

2.2 Organisation du site et de la base de donnée

2.2.1 Base de donnée

En ce qui concerne la base de donnée, nous disposons de 5 tables :

utilisateurs : elle contient toutes les données classiques qui sont nécessaires pour enregistrer un utilisateur. C'est dans cette table qu'on fait la distinction entre les superutilisateurs, les administrateurs et les utilisateurs lampda (variable droit).

trajet : elle contient tous les trajets qui vont être proposés/demandés par les utilisateurs (connectés).

pwdreset : c'est la table qui est utilisée lors de la modification du mot de passe oublié.

réservation elle contient les données liées aux réservations/demandes de trajet réalisé par les utilisateurs (connectés).

commentaire : elle recense tous les commentaires postés sur les pages de réservation de trajet.

Cette base de donnée permet donc le bon fonctionnement du site.

2.2.2 Organisation du site

Nous allons ici détailler l'organisation structurelle du site. Nous détaillerons les choix, erreurs dans une section future. Nous avons créé au total 7 pages accessible par les utilisateurs différentes que voici :

index : il s'agit de la page d'accueil du site qui donne une présentation générale de celui-ci.

proposer un trajet : cette page permet à un utilisateur de proposer un trajet qu'il va réaliser aux autres utilisateurs.

demande un trajet : cette page permet de demander à un utilisateur de réaliser un trajet

voire trajets : on peut regarder les trajets proposés et demandés. Si un trajet nous interesse, on peut alors se rediriger vers une page de réservation.

confirmation de trajet : ici on peut alors choisir le nombre de places que l'on réserve, interagir avec le créateur de l'annonce via les commentaires et bien sur confirmer qu'on réserve ce trajet.

mon compte : cette page permet aux utilisateurs de gérer leurs informations, d'avoir un historique des trajets qu'ils ont proposés et de confirmer les trajets auxquels ils ont participé.

Comme pour la BDD, des idées sont née au cours de la réalisation du projet et c'est pourquoi le rendu final diffère quelque peu de ce qui était prévu dans le cahier des charges.

Nous avons défini ci-dessus les bases du site internet. Maintenant que nous avons une vue globale du projet, nous pouvons alors entrer plus en détail dans les choix réalisés.

2.2.3 Organisation des fichiers

En ce qui concerne l'organisation des fichiers du projet, nous avons choisi d'associer à différents dossiers certaines "fonctions" ;

dossier js : il s'agit du dossier java script. Ce dernier contient les fichiers écrits en JS pour le projet.

dossier image : ce dossier contient les différentes images du projet ; il contient lui-même un dossier membre qui recense les photos de profil des utilisateurs inscrivent.

dossier van : il contient les fichiers qui requièrent plutôt de l'affichage des différentes pages du site.

dossier style ce fichier contient les dossiers de mise en forme des pages (Css).

dossier includes il s'agit dans ce dossier de fichier de traitement de formulaire, ajout/modification de données dans la BDD.

3 Détail des choix réalisés

Expliquons maintenant les choix qui nous ont mené à ce résultat.

3.1 Page Index

3.1.1 Header

Avant de parler de la page d'index, nous allons rapidement présenter le header, qui est présent sur la totalité du site. Cette partie du site, primordiale, permet de naviguer sur les différentes pages citées plus haut. Le header prend en compte l'état **connecté/deconnecté** des utilisateurs, que nous détaillerons dans la section de la page **index**. L'aspect pratique est qu'il permet grâce à la méthode **SESSION** d'avoir accès à **l'id de l'utilisateur connecté** sur n'importe quelle page (et d'autres données par ailleurs). En effet, on "lance une session" avec **session_start** et on test si l'utilisateur est connecté où non. En fonction de ces tests, on affiche dans le header les données correspondantes.

3.1.2 Index

Il s'agit de la page d'accueil du site. Elle détaille comme expliquait l'objectif du site. Il s'agit de **la seule** page (hors connexion et inscription) qui est accessible lorsqu'un utilisateur n'est pas connecté/n'a pas de compte. Nous avons pris le parti de rendre accessibles les fonctionnalités de notre site seulement aux utilisateurs recensés sur celui-ci. En effet, chaque action requiert que l'utilisateur se soit enregistré dans la BDD. Ainsi on distingue deux fonctionnements :

3.1.3 Connexion

La page de connexion est une page clé du projet. Elle permet à un utilisateur de se répertorier parmi nos membres et d'avoir accès à notre site. L'utilisateur aura à remplir un formulaire classique dans lequel il devrait renseigner ses informations personnelles. L'ajout d'une photo de profil reste optionnel.

3.2 Modification du mot de passe

Lorsqu'un utilisateur désire changer son mot de passe, nous créons deux tokens associés à sa demande : un token temporel et un token d'identification. Le token temporel permet de rendre la demande éphémère. De plus, le token d'identification permet de rendre le changement de mot de passe sécurisé. En effet, l'utilisateur n'étant pas connecté pour changer le mot de passe, il a besoin du token de sécurité pour changer de mot de passe. Celui-ci est récupéré dans le lien envoyé par mail à l'utilisateur,

ce qu'on considère sécurisé pour l'utilisateur. De ce fait seul lui aura accès au changement de mot de passe car le token n'est pas devinable.

- l'utilisateur n'est pas connecté, alors il peut voir la page de présentation sur laquelle il trouvera la redirection vers le formulaire d'inscription ou la page de connexion. De plus, il trouvera également ces redirections dans l'header. Nous avons pour cela utilisé seulement des liens de redirection (balise `<a>`) associée à des boutons (balise `button`).
- l'utilisateur est connecté, auquel cas il peut accéder aux fonctionnalités du site.

Pour ajouter un côté ludique, nous avons ajouté la possibilité de voir sur une map la ville concernée par le site internet. Pour cela il faut seulement créer le lien d'affichage grâce à l'API de Google. Ensuite ont utilisé une balise `<iframe>` pour aller lire le lien qu'on a généré.

3.3 Proposer un Trajet

Cette partie est dédiée aux utilisateurs qui souhaite partager un voyage avec les villageois. Pour cela il va remplir un formulaire en entrant les données de la ville de départ, d'arrivée ainsi que les dates du trajet. Une fois le trajet validé, il sera entré en base de données avec le type proposé pour le dissocier d'un trajet demandé. Cette fois, pour que l'utilisateur puisse vérifier qu'il ne s'est pas trompé dans les coordonnées rentrées, il peut regarder sur la carte les marqueurs, qui pointent sur les localisations choisies. Pour cela, on a utilisé l'API de Leaflet, qui permet d'intégrer des cartes à un site internet. Pour réaliser l'affichage de ces pointeurs, on a utilisé des écouteurs placé sur les input des villes. Ainsi, lorsqu'on quitte ces champs, a créé un pointeur sur la carte à l'endroit entrée. Pour cela, dans un fichier JS, ont récupéré les données entrées, puis à l'aide de l'API Nominatim/Openstreetmap on récupère les coordonnées Gps de l'emplacement (requête JSON). Ensuite, on ajoute le pointeur aux bonnes coordonnées.

Pour le formulaire, on procède à un traitement comme pour l'inscription.

Axe d'amélioration : Il serait intéressant de pouvoir utiliser la technique de drag sur la carte. Pour cela, il faudrait faire le chemin inverse en allant chercher l'adresse en fonction des coordonnées Gps, puis mettre à jour le formulaire. Une seconde amélioration serait d'afficher le trajet Gps entre les deux pointeurs.

3.4 Demander trajet

pour cette page, le fonctionnement est similaire. Lors de l'entrée en BDD, on met simplement un type demande pour distinguer le trajet de la section précédente.

3.5 Voir Trajets

Dans cette partie, on peut comme son nom l'indique voir quels trajets ont été proposés/demandés par les autres utilisateurs. On dispose donc de deux tableaux distincts. L'un contient les demandes et l'autre les propositions. Dans ces tableaux, les affichages se font par ordre de création dans la page Proposer trajet. (on utilise un "ordre by" id de trajet). En effet, chaque trajet possède un identifiant unique qui s'incrémente au fur et à mesure des créations. Ainsi, on n'est pas sûr d'avoir cet ordre. (un autre ordre d'affichage aurait pu être par date). Dans les tableaux, chaque trajet est affiché sous forme de vignettes, qui détiennent toutes les informations utiles au trajet. Lorsqu'un utilisateur est intéressé par un trajet, il peut alors se rendre sur la page de confirmation de trajet, afin de réserver le trajet. Pour réaliser cet affichage, on utilise du Php en faisant une requête SQL à la base de données. Une fois la requête exécutée, on place les résultats dans un tableau (`fetch All()`). On mélange ensuite balises HTML et Php pour réaliser l'affichage en sélectionnant chaque ligne du tableau (`foreach`). Ensuite, la mise en forme se fait en CSS en donnant des styles aux différentes divisions/boutons.

Axe d'amélioration : Ici, l'amélioration se fera principalement au niveau du design, en ajoutant pourquoi pas du JS pour rendre l'UX plus agréable.

3.6 Confirmation de trajet

Axe d'amélioration : Ici, l'amélioration se fera principalement au niveau du design, en ajoutant pourquoi pas du JS pour rendre l'UX plus agréable.

3.7 Confirmation de trajet

Cette page est divisée en deux fonctions principales. D'un côté on peut interagir avec le créateur de l'annonce à l'aide de l'espace commentaire et d'un autre on peut réserver le trajet.

pour ce qui est de la partie commentaire, il s'agit d'un ajout qui n'était pas prévu au départ. Nous avons pensé qu'il serait intéressant pour les utilisateurs de pouvoir interagir avec la personne qui a publié l'annonce. En effet, il peut demander de décaler l'heure du départ, ou simplement si l'annonce est toujours valide. Pour cette partie, on a utilisé du Php. Lorsqu'un commentaire est publié, on le rentre en BDD, puis lorsque la page s'actualise, on affiche tous les commentaires liés à ce trajet dans l'espace dédié à l'aide d'une requête SQL.

pour ce qui est de la réservation, l'utilisateur peut choisir le nombre de places qu'il réserve. Une fois choisi, il peut alors réserver son trajet. Nous avons décidé ici de stocker en BDD le nombre de places réservées afin de pouvoir le mettre à jour lorsqu'une réservation est faite. En effet, lorsqu'un prochain utilisateur voudra réserver ce même trajet, il ne pourra alors choisir que le nombre de places qui reste. Une fois qu'il n'y a plus de place à réserver le trajet disparaît de la page voir Trajet. De même on a empêché un utilisateur de réserver son propre trajet (option disable associée à du Php). Enfin on a ajouté l'envoyé automatique d'un mail avec un code de validation qui sera à entrer par le conducteur une fois le trajet réalisé.

Axe d'amélioration : Pareil pour cette page, ajouter du JS pour permettre de faire toutes les actions depuis la page précédente. Il faudra donc repenser le design.

3.8 mon compte

sur cette page, nous avons fait le choix de créer un système d'onglet afin de dynamiser le parcours entre les différentes parties. Ici on a utilisé du JS pour cela. Détaillons les sections :

mon compte : cet onglet permet à l'utilisateur de consulter ses données et, s'il le souhaite, de les modifier. Dans secas, il sera redirigé vers une page dédiée à cet effet. Il pourra alors modifier toutes les données comme il le souhaite. En ce qui concerne sa photo de profil (s'il en a rentré une), il pourra la supprimer et la modifier.

mes trajets : il s'agit ici d'afficher les trajets qui ont été proposé par l'utilisateur en question. Pour rester cohérent, nous avons gardé l'aspect vignette de la page confirmer trajet. On a également ajouté la possibilité de modifier le nombre de places proposé par l'utilisateur, ou alors de carrément supprimer le trajet (suppression dans la bdd). Nous avons ici choisi de ne pas laisser la possibilité à l'utilisateur de modifier la destination de départ ou d'arrivée. Nous préférons qu'il supprime le trajet et qu'il en crée un nouveau.

Confirmer trajet : cette section est dédiée au conducteur ayant réalisé un trajet. L'objectif est qu'une fois que le trajet a eu lieu, le passager envoie au conducteur le code qu'il a reçu dans ses mails. Le conducteur devra alors l'entrer dans cette partie dans le trajet correspondant. Une fois fait, on mettra à jour la variable nombre_place_validée de la BDD. Celle-ci sert à compter un nombre de points obtenus par un conducteur à force de réaliser des trajets(visible dans mon compte).

Administrateur : ici la partie est dédiée aux mairies. En effet elles posséderont un accès administrateur qui leur permettra, via cet onglet, de visionner tous les trajets qui ont eu lieu. Elles pourront voir, modifier et supprimer les informations des utilisateurs (sauf leur droit et leur id). Nous souhaitons améliorer l'espace administrateur au niveau de la gestion des trajets/trajets réservés.

Super-Administrateur : cette section est réservée aux créateurs du site. Il permet en plus des fonctionnalités de l'espace administrateur de modifier les droits de tous les utilisateurs.

3.9 L'UX

Nous avons cherché à avoir la meilleure expérience utilisateur possible étant donné que notre application est destinée au public. Pour cela nous avons cherché un design simple et épuré sans animation superflue et avec des boutons simples et clairs. Nous avons cherché à avoir un design cohérent dans l'ensemble. Nous avons rendu les pages accessibles aux utilisateurs responsive en utilisant le système de grille de bootstrap. Le header est également responsive et se réduit à un menu déroulant pour les appareils au petit format.

3.10 La sécurité

Bien que nous ne soyons pas expérimentés en sécurité, nous avons essayé de prendre soin aux méthodes que nous utilisons dans les formulaires. Nous avons privilégié la méthode "POST" et nous avons pris soin de vérifier à chaque page "includes" que l'utilisateur y avait accès via le bouton dédié.

Axe d'amélioration : Pour cette partie, il faut dans un premier temps revoir le design de la page confirmer trajet qui reste trop simple et qui n'est pas en faveur de l'UX. Dans l'onglet trajet, il faudrait ajouter une distinction entre trajet proposé et demandé afin de permettre à l'utilisateur de s'y retrouver clairement.

4 La suite du Projet

Pour la suite du projet, nous souhaitons poursuivre la réalisation de projet au C19 lors de notre stage. Nous avons déjà quelques idées pour améliorer l'existant comme :

- Une messagerie privée avec notifications rendrait également l'UX beaucoup plus agréable car pour le moment les utilisateurs doivent se rendre dans les pages qu'ils ont commenté pour voir s'ils ont eu une réponse ou non.
- Autres axes déjà cités.

5 Conclusion

En conclusion le projet a été très enrichissant pour apprendre à travailler en équipe. Il fallait tous s'accorder pour parvenir au rendu attendu, malgré la difficulté liée notamment à la distance.

Enfin il semblerait qu'il nous manque de l'expérience en CSS/JS pour rendre l'UX plus agréable et professionnelle. À l'avenir nous comptons reprendre ce site web avec différents frameworks pour coder de manière plus efficace.

6 Annexe

Voici le schéma de BDD

