

Compte Rendu de TP2 de Systèmes Répartis (8INF843)

Auteurs : Sébastien CORFA / Camille CROS / Vincent PERA

Nous avons choisi de réaliser le premier exercice proposé dont la consigne était la suivante :

« Afin de contribuer à la protection de l'environnement, une entreprise TravelExpress.com (semblable à amigoexpress : <http://www.amigoexpress.com/>) désire développer une application distribuée (Web) de covoiturage. L'application permettra aux gens qui partagent les mêmes parcours de se connaître et de s'organiser pour faire du covoiturage. »

Le site web est accessible à l'adresse suivante : <http://51.255.41.18/CovoitExpress/>
Un compte de test est disponible :

- Identifiant : **test**
- Password : **test**

D'autre part, le code source du site web est disponible sur Github à l'adresse suivante : <https://github.com/sebspas/CovoitExpress>

A. Introduction et description du système

a. Technologies utilisées

Nous utilisons une base de données phpMyAdmin présente sur le serveur VPS (chez OVH) de Sébastien CORFA. Le schéma de base de données a été réalisé via le logiciel MySQL Workbench.

Le contrôleur et le modèle sont programmés en PHP et la vue est programmée en Twig.

De plus, WampServer a été utilisé pour programmer de manière locale sur le serveur. Nous avons pu utiliser la base de données locale de WampServer dans les cas où la base de données distante n'était pas disponible, comme dans les locaux de l'UQAC par exemple.

Nous avons utilisé le logiciel Composer, qui permet de gérer facilement les dépendances avec les bibliothèques externes, telles que Twig dans notre cas.

Enfin, nous avons utilisé PHPStorm et Visual Studio Code comme IDE pour le développement PHP / HTML / CSS / JS / Ajax / Twig. Les deux IDE étant optimisées pour travailler sur le PHP et le code côté serveur (surtout PHPStorm).

b. Descriptif du système

Notre application web se divise en 5 pages accessibles par l'utilisateur :

1. La **page de connexion** où l'utilisateur a le choix entre créer un compte pour pouvoir accéder au reste du site ou se connecter directement s'il a déjà un compte existant.
2. La **page d'accueil** (onglet "**Home**") sur laquelle sont recensés tous les trajets disponibles pour l'utilisateur. Les trajets sont récupérés depuis la base de données distante via requête SQL. Les trajets créés par l'utilisateur et dont la date de début est déjà dépassée ne sont pas affichés. L'utilisateur peut également effectuer une recherche de trajet selon différents critères et ordres d'affichage (ex. prix décroissants). Le tri des trajets est affiché dynamiquement sur la page.
3. La **page de création de trajets** (onglet "**Create a travel**") depuis laquelle l'utilisateur peut proposer un nouveau trajet. Le trajet sera ajouté à la base de données s'il est considéré comme valide. Pour que le trajet soit valide, l'utilisateur doit impérativement indiquer les informations suivantes :
 - 3.1. La ville de départ et la ville d'arrivée
 - 3.2. La date et l'heure de départ
 - 3.3. La date et l'heure d'arrivée
 - 3.4. Sa marque d'automobile
 - 3.5. Le nombre de places disponibles
 - 3.6. Le prix qu'il veut fixer pour ce trajet
4. La **page de profil** (onglet "**Profile**") où l'utilisateur peut retrouver toutes les informations en lien avec son compte. La page contient donc 3 blocs d'informations :
 - 4.1. Le bloc de gauche contient :
 - 4.1.1. L'avatar, le pseudo et le solde courant
 - 4.1.2. La rubrique "Your information" avec la ville d'habitation, l'e-mail et le numéro de téléphone

- 4.1.3. La rubrique "Your preferences" avec les préférences pendant le trajet (ex: "No Fast Food"). Ces préférences peuvent être modifiées par l'utilisateur depuis cette page : ouverture d'une "modal" pour l'ajout ou la suppression d'une préférence parmi celles disponibles dans la base de données. Nous utilisons de l'Ajax pour gérer cette modification et pour mettre à jour en temps réel l'affichage des préférences sans avoir à rafraîchir la page manuellement.
- 4.2. Le premier bloc au centre de la page est un historique des trajets qui ont été proposés par l'utilisateur. Chaque trajet est identifié par :
 - 4.2.1. Sa ville départ et sa ville d'arrivée
 - 4.2.2. Son horaire de départ (date et heure)
 - 4.2.3. Son prix
 - 4.2.4. Un lien vers la page du trajet
- 4.3. Le second bloc au centre de la page est un historique des trajets auxquels l'utilisateur a participé en tant que passager. Il recense le même type d'information que le premier bloc. Les trajets affichés dans ces deux blocs sont récupérés depuis la base de données distante via requêtes SQL.
- 5. La **page du trajet** (accessible depuis les onglets "**Home**" et "**Profile**") permet à l'utilisateur d'avoir une vision globale de toutes les informations relatives au trajet. Cette page contient deux blocs d'informations :
 - 5.1. Le bloc de gauche recense toutes les informations sur le trajet et sur le conducteur dudit trajet. L'utilisateur peut également réserver sa place en appuyant sur "Buy a seat" placé en bas de ce bloc.
 - 5.2. Dans le bloc de droite, nous avons inséré une carte du trajet proposée par Google Maps. Cette carte permet de donner à l'utilisateur une estimation du temps de trajet et de la route qui pourra être empruntée.

B. Schémas et diagrammes de conception

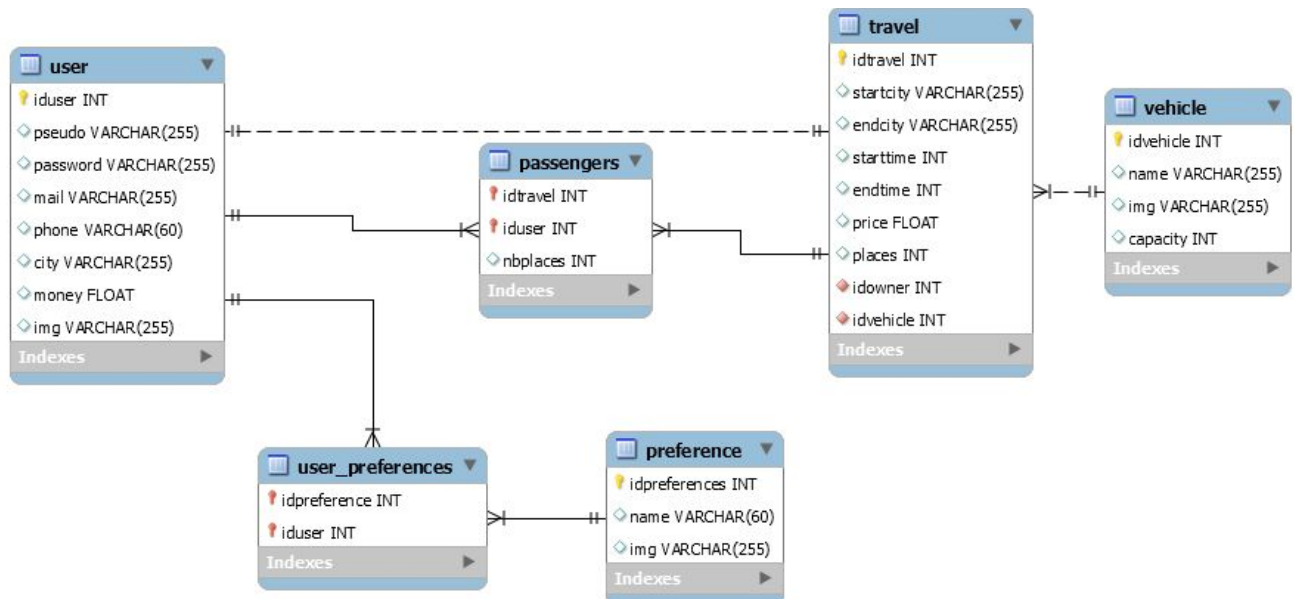


Schéma de la base de données

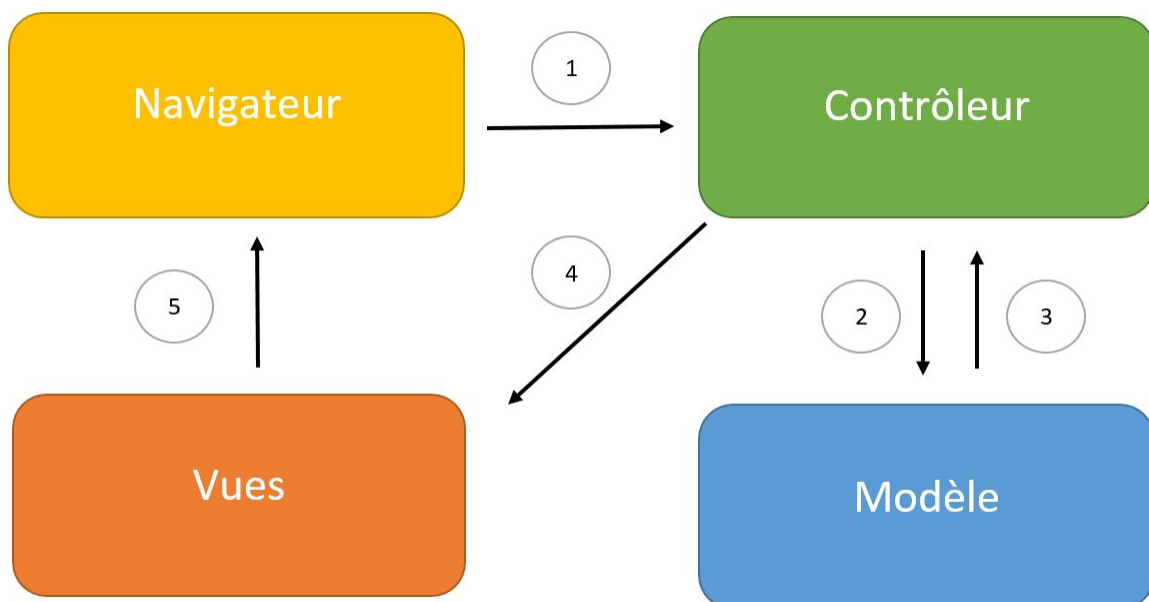


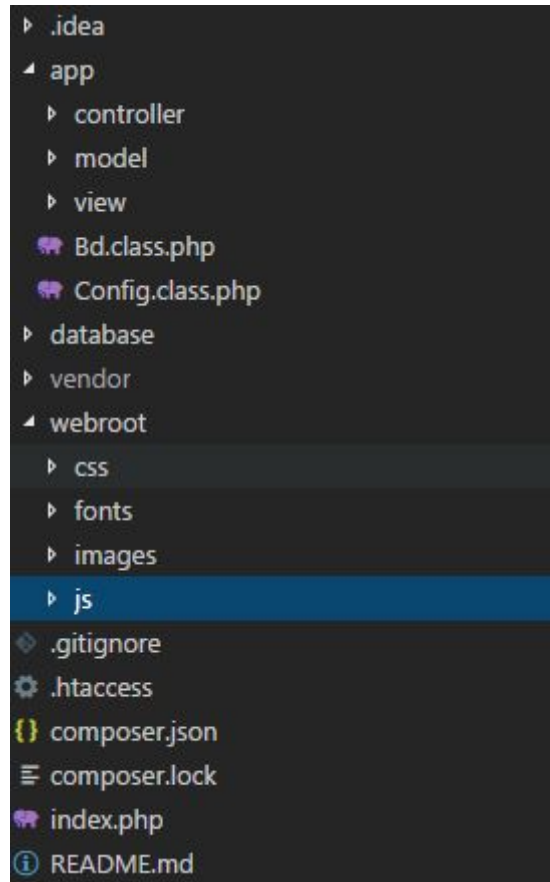
Schéma de conception

Nous avons décidé d'implémenter la partie serveur selon la patron de conception MVC (Model-View-Controller).

Dans un premier temps, le navigateur du client appelle le contrôleur (en PHP) de la page qu'il souhaite obtenir.

Le contrôleur va importer le modèle correspondant pour l'utiliser dans la conception de la page. Le modèle, qui est aussi en PHP, va en effet réaliser toutes les requêtes base de données nécessaires à l'action choisie par l'utilisateur et à l'affichage de la vue suivante.

Le contrôleur va ensuite appeler la vue qui est en Twig. La vue va recevoir les informations du contrôleur et les mettre correctement en forme dans un format HTML qui est renvoyé au navigateur de l'utilisateur.



Hiérarchie du Projet

Dans la hiérarchie du projet, le dossier app contient notre modèle MVC ainsi que le fichier Config et Bd, le premier servant à la configuration à la connexion à la base de données distante ou locale, le second permettant de se connecter aux tables et contenant des fonctions de base en SQL comme Select, Update ou Delete.

Le dossier webroot contient tous les éléments utiles au côté client, comme les CSS, images de base ou encore des fichiers JavaScript.

Le dossier database contient le fichier utilisé dans MySQL Workbench ainsi que l'image visuelle de la base de données telle que vue précédemment dans ce rapport.

Le dossier `.idea` étant généré par PHPStorm lors d'un projet, celui-ci concerne uniquement l'IDE.

Le dossier `vendor` est utilisé par Composer pour gérer les dépendances avec les librairies extérieures, comme nous en avons eu besoin ici avec Twig.

C. Le guide de l'utilisateur

Lorsque l'usage arrive sur la plateforme, il peut se connecter s'il possède déjà des identifiants, sinon il peut se créer un compte en cliquant sur le bouton "SIGN UP".

Dans le cas de la création d'un compte, l'utilisateur devra renseigner différents éléments comme son mot de passe, un identifiant, une adresse email valide ou encore sa ville de résidence ou son téléphone.

Sur la page d'Accueil (onglet "Home"), on peut apercevoir les trajets disponibles dans l'application. Grâce à la console de recherche à gauche, on peut trier les trajets suivant les villes de départ ou destination, la date de départ, le prix ou encore le nombre de places disponibles. On peut aussi trier les résultats par prix croissant ou décroissant, par date ou par nombre de places.

Depuis la page d'Accueil également, vous pouvez cliquer pour réserver un trajet. Vous arriverez alors sur la page du trajet détaillée. En cliquant sur le bouton de réservation, vous devrez choisir le nombre de sièges à acheter et ensuite valider la commande.

Si la réservation est possible, elle est disponible depuis votre page profil (onglet "Profile").

En cliquant sur "Créer un trajet" (onglet "Create a travel") en haut à droite, on arrive sur une interface dédiée où le conducteur doit remplir les données de son trajet. Une fois saisi et envoyé, le trajet est disponible depuis le profil de l'utilisateur dans la catégorie "Voyages en tant que Conducteur" (aka "Trips as driver").

En bas à gauche, vous pourrez éditer vos préférences afin d'avertir vos futurs passagers de ce que vous préférez, comme si vous n'acceptez pas que les gens fument dans votre voiture par exemple.

Enfin, on peut se déconnecter à tout moment en cliquant sur le bouton de déconnexion tout à droite de la barre de navigation. On serait alors redirigé vers la page de connexion.

D. Les traces d'exécution

Pour récupérer des traces d'exécution, nous avons choisi de vous présenter la série de captures d'écran ci-dessous. Il s'agit ici de montrer tous les types de pages et autres feedbacks visuels auxquels l'utilisateur sera et/ou pourra être confronter au cours de son utilisation. Nous nous sommes aussi attardés à mettre en avant l'interaction avec notre base de données distante (ajout d'un utilisateur dans la base de données, modification des préférences de l'utilisateur, etc.).

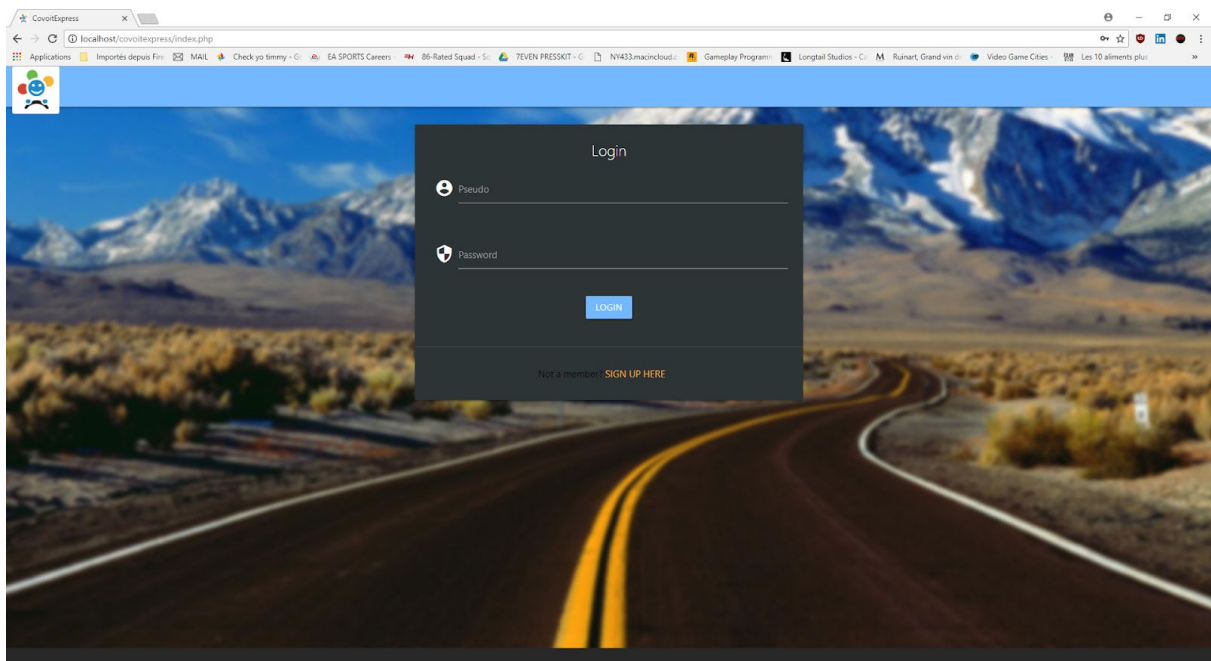


Image 1 : Interface de Connexion

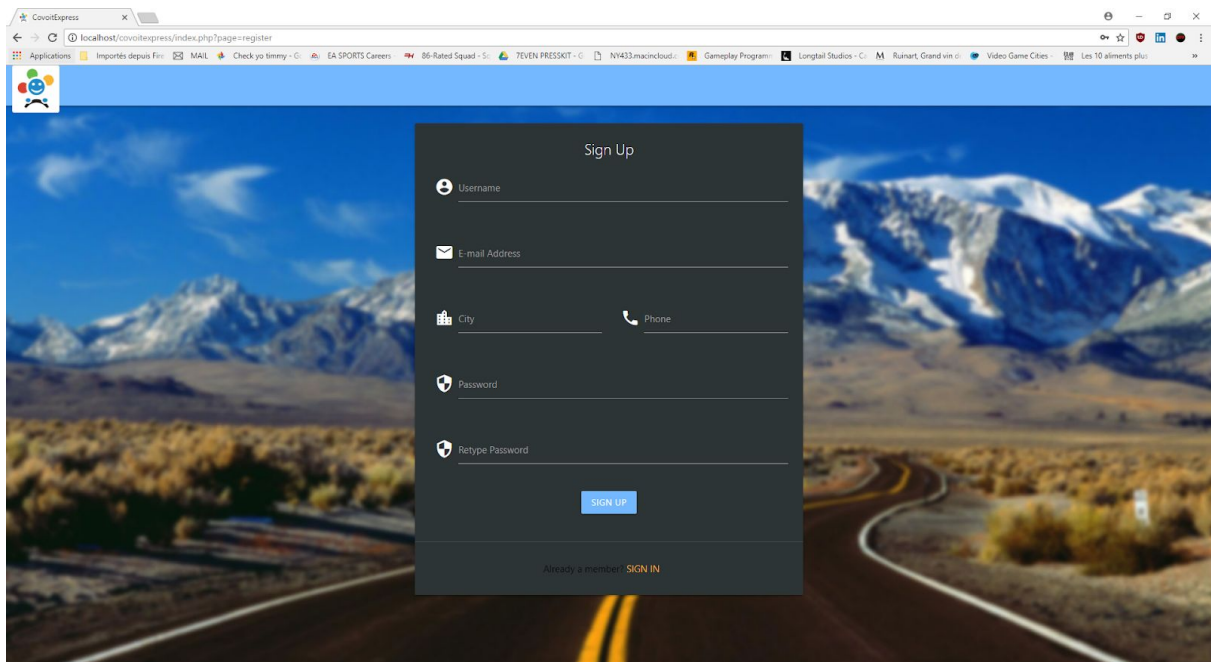


Image 2 : Interface d'Inscription

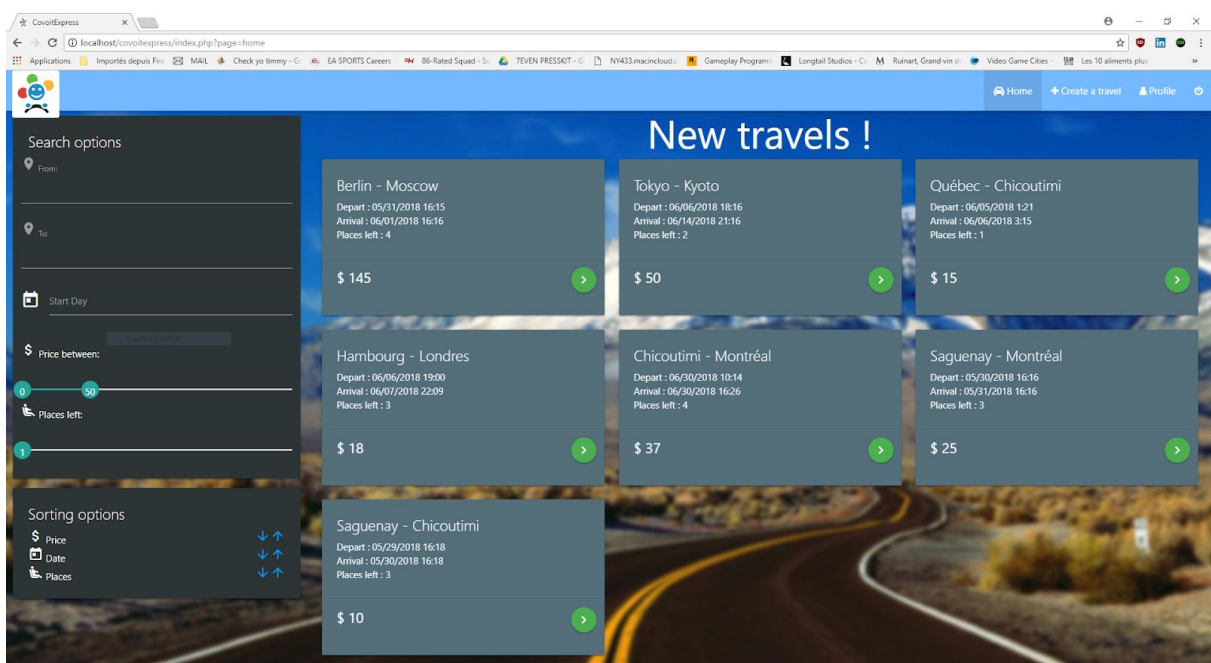


Image 3 : Page d'Accueil

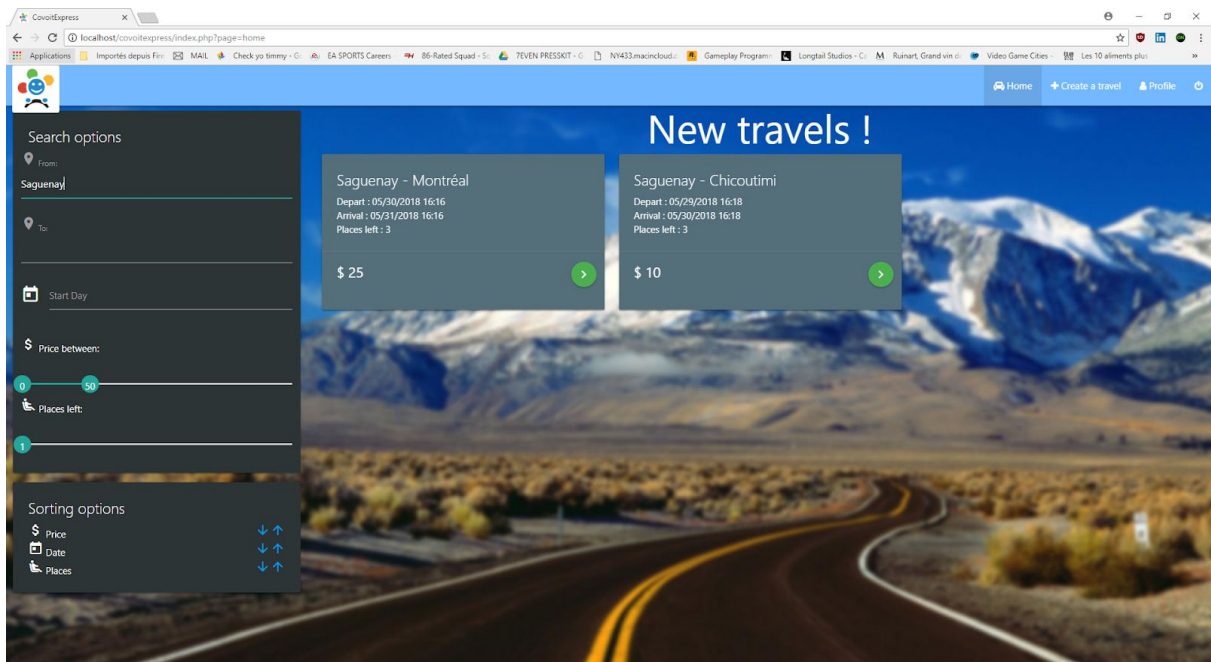


Image 4 : Recherche via Ville de Départ

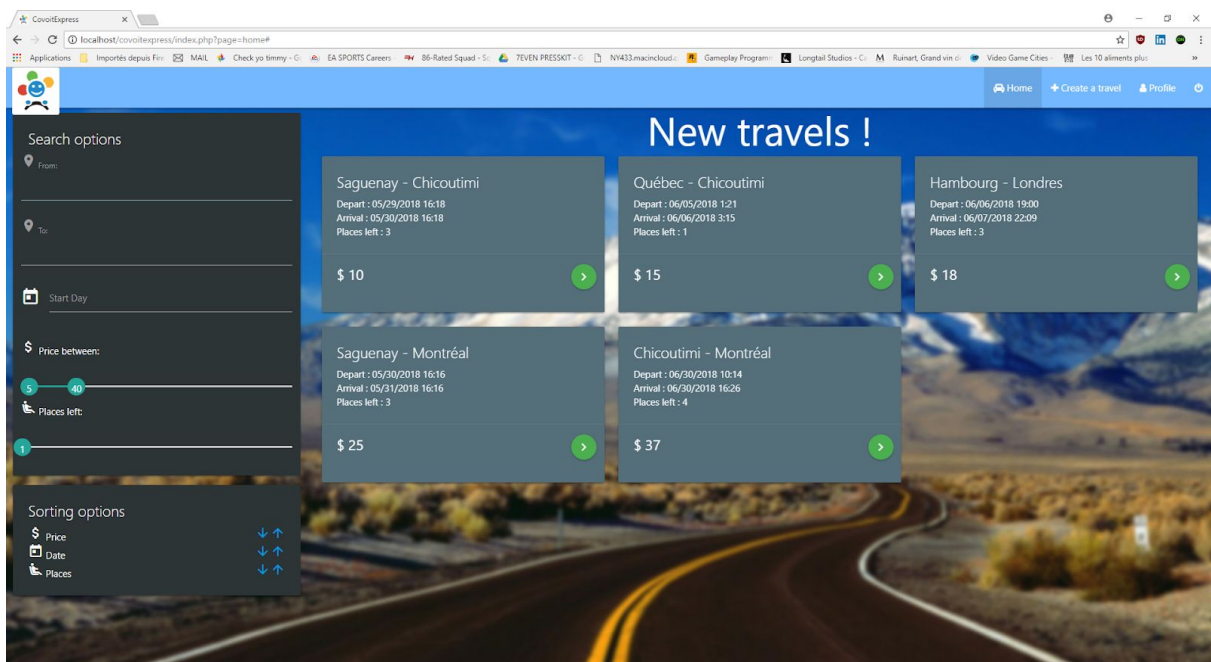


Image 5 : Recherche via Prix trié en ordre croissant

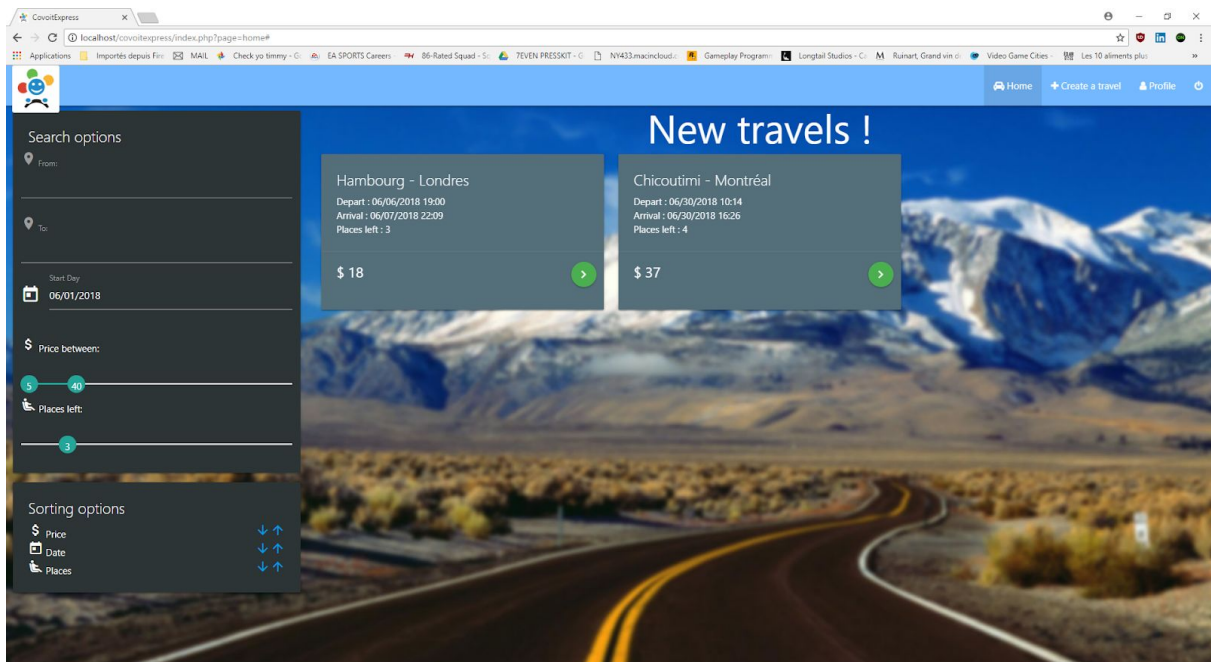


Image 6 : Recherche via Prix et Nombre de Places, trié par le Prix en ordre croissant

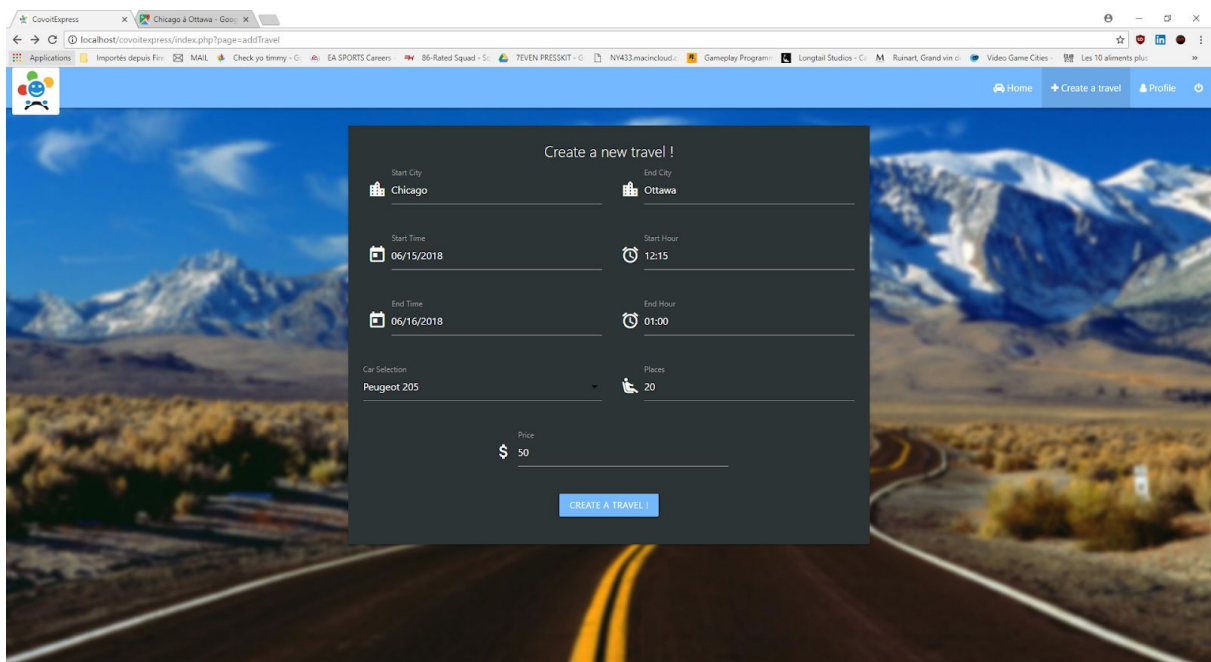


Image 7 : Ajout d'un trajet

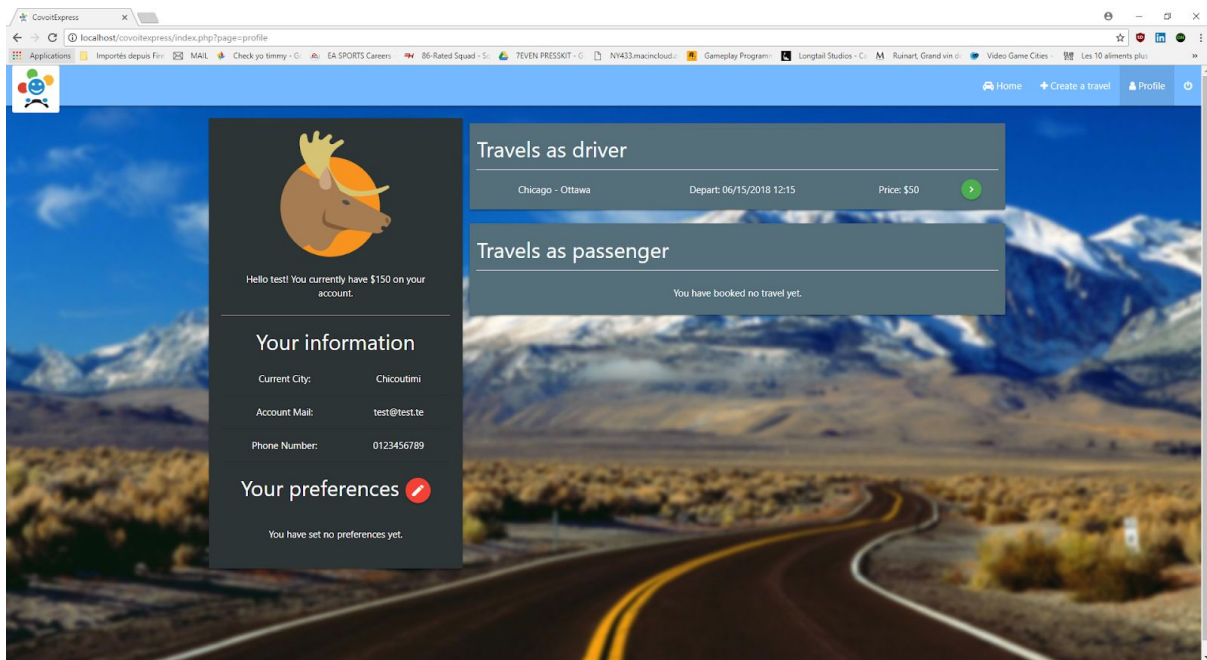


Image 8 : Vue du profil

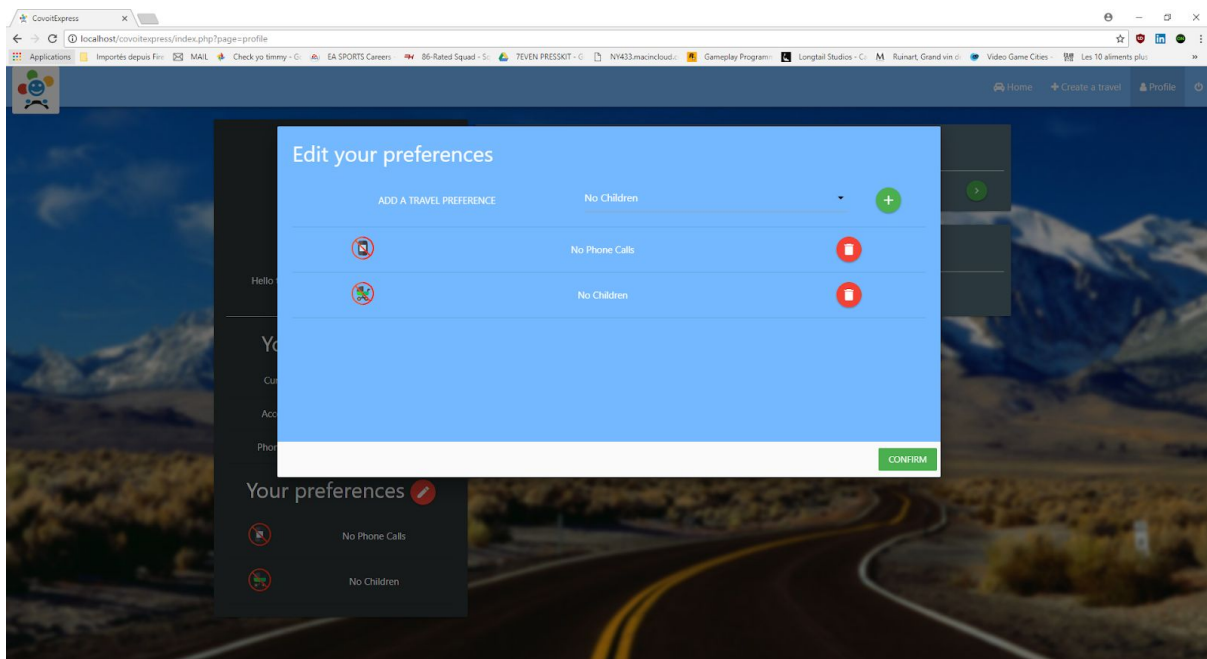


Image 9 : Ajout des préférences à son profil

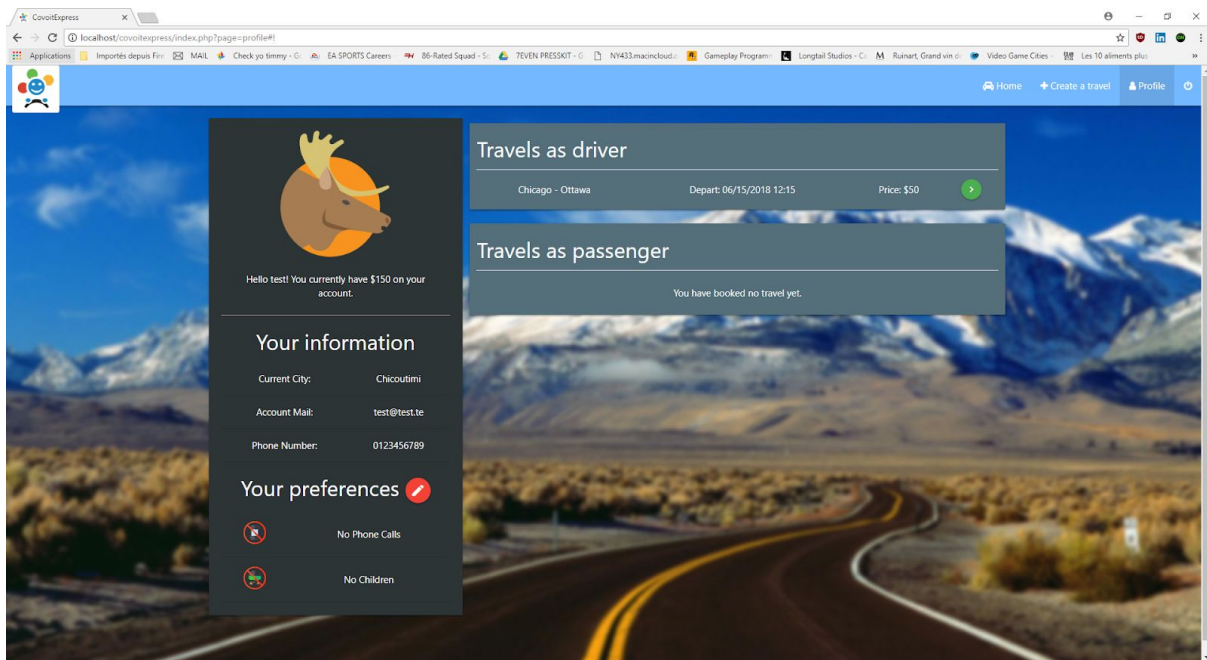


Image 10 : Vue du profil après ajout des préférences

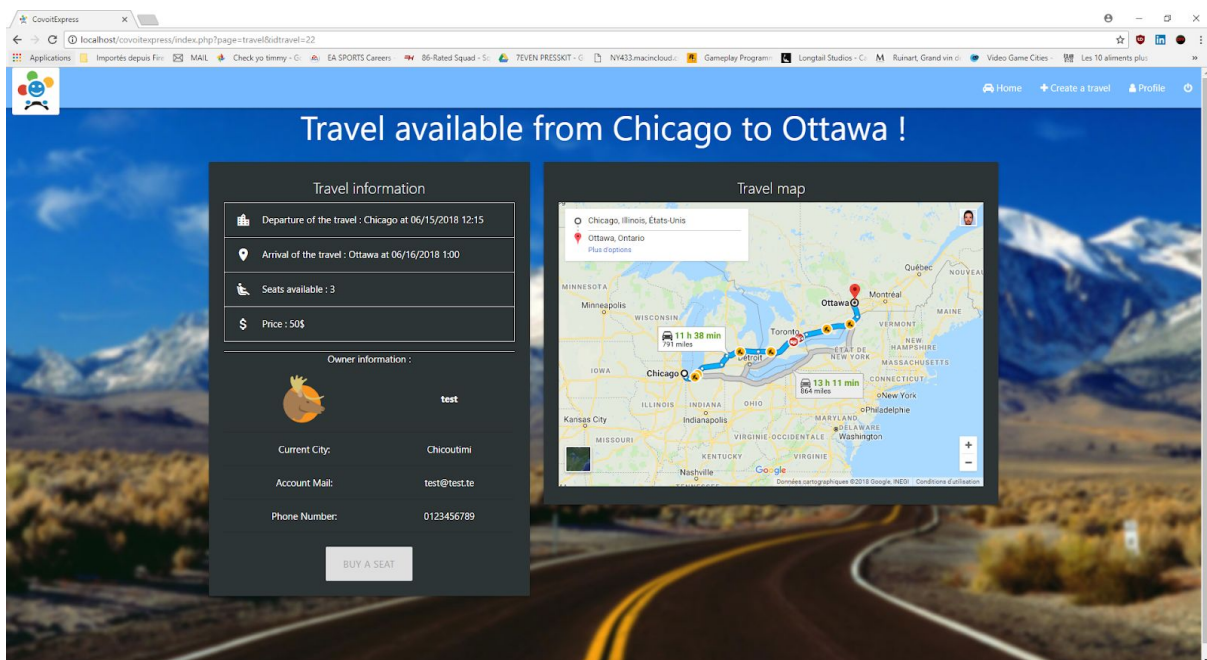


Image 11 : Accès au trajet que l'on propose via notre profil

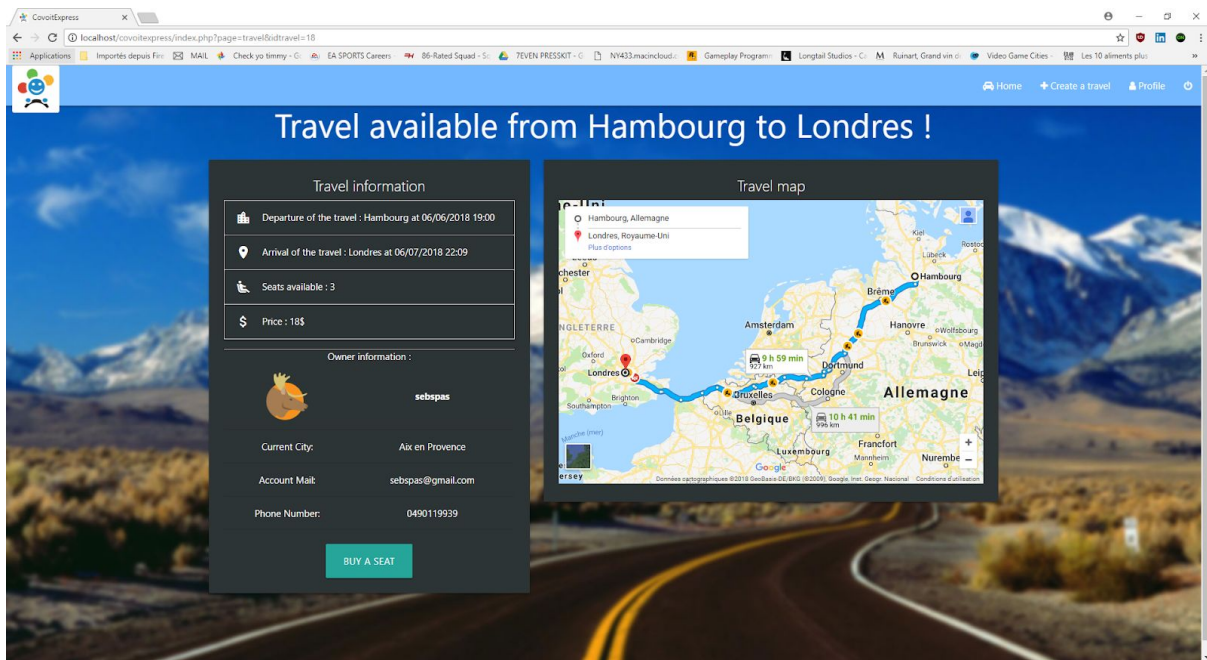


Image 12 : Accès aux détails d'un trajet via la recherche de la page d'Accueil

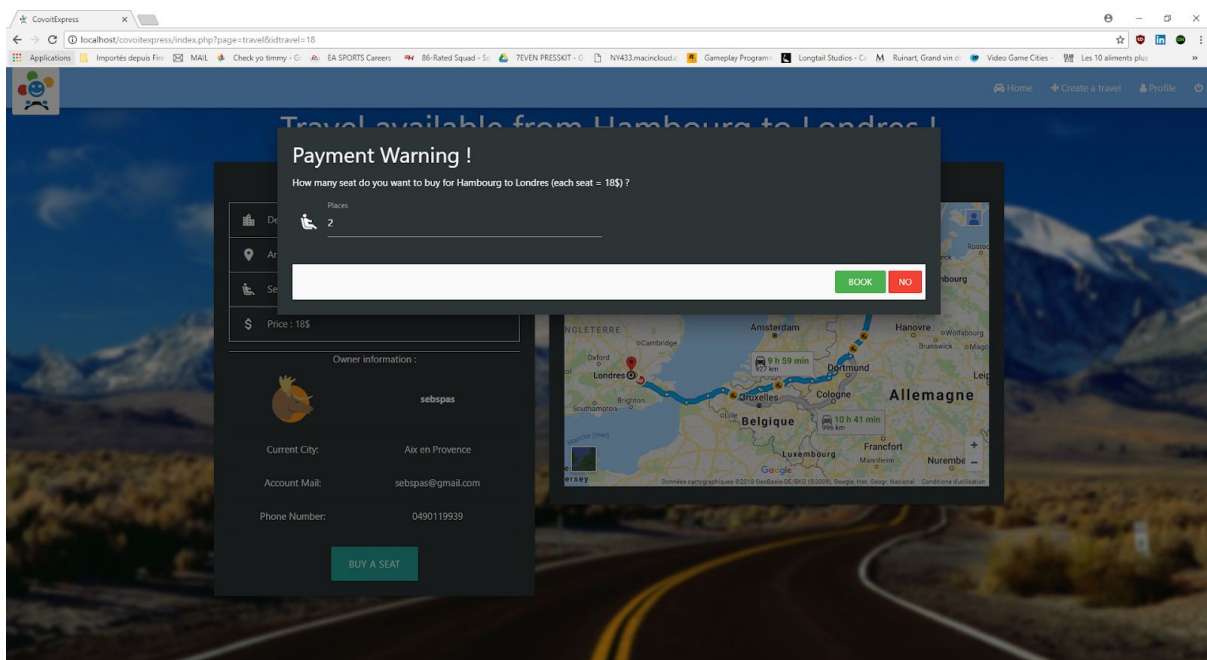


Image 13 : Réservation de sièges

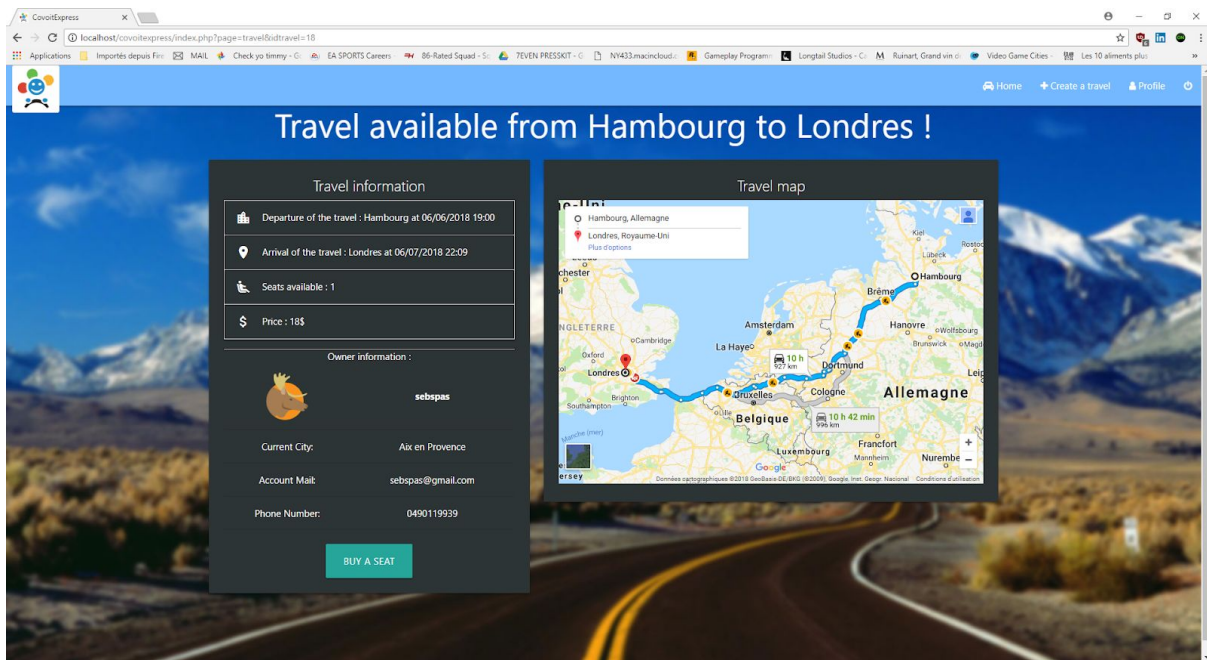


Image 14 : Interface mise à jour après la réservation de 2 sièges

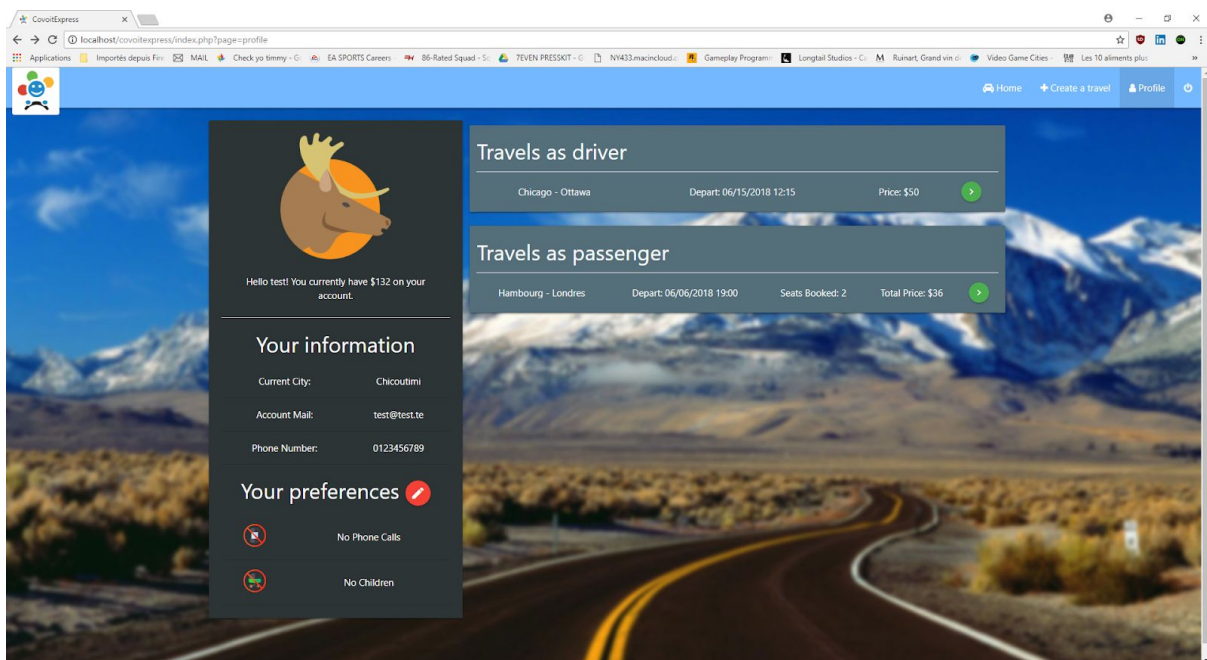


Image 15 : Profil mis à jour après la réservation