

Compte-Rendu PWEB:

Ressources utilisées:

https://www.youtube.com/watch?v=_WS5ZNI043s (tuto symfony 10 épisodes ~ env 15H)
<https://symfony.com/doc/current/index.html> (documentation de symfony)

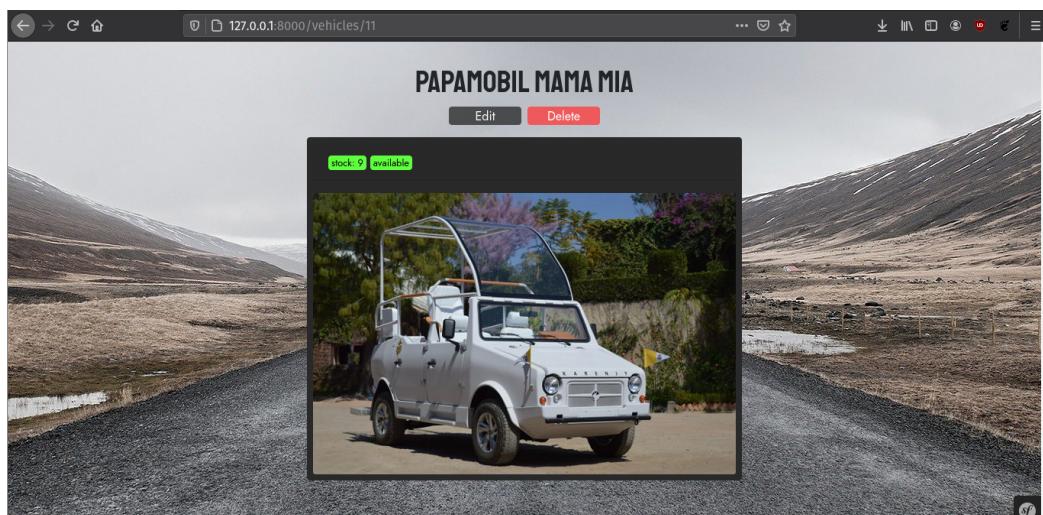
État des avancements :

Nous avons essayé de réaliser un MVP (minimum viable product) en nous basant sur le plus de framework et librairies possibles afin d'accélérer le développement, tels que Symfony 5 et Bootstrap.

Le site présente un header avec un menu, d'une page principale correspondant au service d'affichage de tous les véhicules, et d'autres routes pour chaque service.

app_home	GET	Any	Any	/
app_vehicles_show	GET	Any	Any	/vehicles/{id}
app_vehicles_edit	GET PUT	Any	Any	/vehicles/{id}/edit
app_vehicles_create	GET POST	Any	Any	/vehicles/create
app_vehicles_delete	DELETE	Any	Any	/vehicles/{id}/delete

Tout ce qui concerne la gestion des véhicules (ajout, suppression, modification, affichage) est fonctionnel.



L'on peut uploader une image par voiture, la prévisualiser, la télécharger et la supprimer.

Ces images sont enregistrées côté serveur et leurs chemins d'accès sont stockés dans l'entité véhicule. Leurs noms sont régénérés pour qu'ils soient uniques. Tout ceci est géré par l'ORM de Symfony appelé Doctrine que nous avons appris pour ce projet.

The image shows a file tree on the left and a web browser window on the right. The file tree displays a directory structure under 'public': 'css' (header.css, styles.css, vehicles.css), 'media / cache', 'pictures', and 'uploads / vehicles' containing several image files. The browser window shows a 'New Vehicle' form with fields for 'Type' (a dropdown menu), 'Image (JPG or PNG file)' (a file input field with 'Browse...' and 'No file selected.' buttons), and 'Stock' (a dropdown menu). The background of the browser window features a landscape image.

Pour l'édition des propriétés JSON, une textarea spéciale en thème sombre avec une police monospace et une fonctionnalité de tabulation permet une édition plus agréable. La syntaxe est vérifiée côté serveur pour ne pas que les utilisateurs fassent planter les fonctions d'affichage en rentrant du mauvais JSON. Ensuite dans l'affichage, un tableau bootstrap est généré.

The image shows two side-by-side views. On the left, a dark-themed textarea labeled 'Characteristics (JSON)' contains JSON code. A red error bar at the top says 'You've entered an invalid Json.' Below it, the JSON code is shown:

```
{---  
    ///"weight":10,  
    "year":2004,  
    "price":"8002€"  
}]]]
```

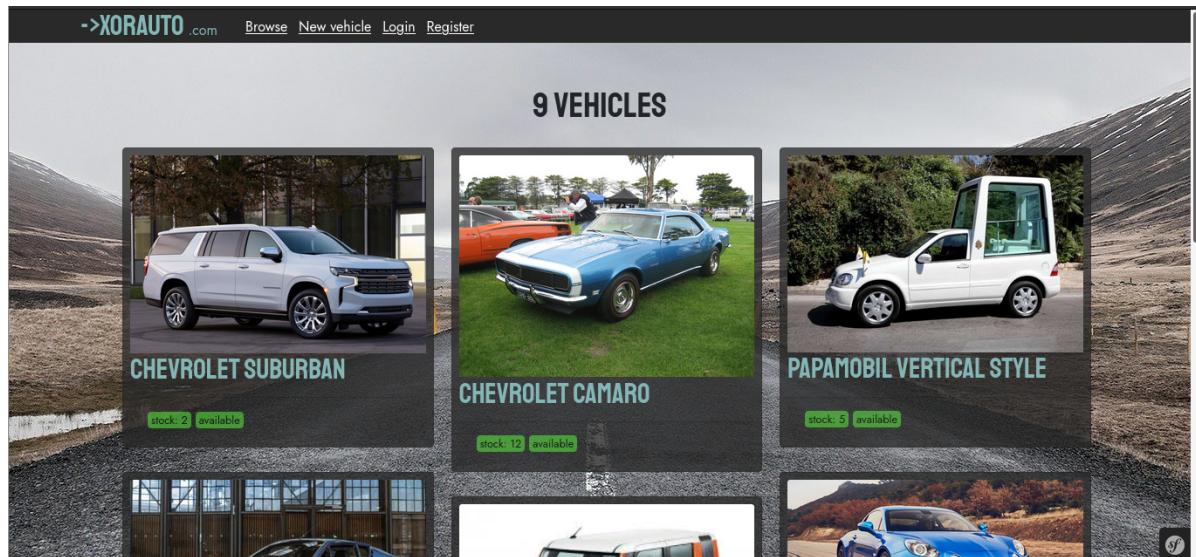
 On the right, a table with a dark header row shows vehicle characteristics: 'stock: 10' and 'available'. The table has two rows: 'poid' with value '700kg' and 'autonomie' with value '500km'.

Les champs sont validés côté serveur (taille min, taille max, validité syntaxique, taille de l'image < 8Mo).

La fonctionnalité de création de compte loueur est en cours de réalisation. Pour l'instant les utilisateurs anonymes ne sont pas limités dans leurs actions. Le site est plutôt joli et totalement responsive, grâce à un mélange entre CSS customisé et bootstrap pour la structure et le responsive. Nous avons utilisé bootstrap grid, bootstrap card et bootstrap table dans le site. Nous avons aussi

The image shows a dark-themed navigation bar with links: 'Browse', 'New vehicle', 'Login', and 'Register'. Below it is a green success message bar: 'Vehicle successfully updated!'. At the bottom center, the text '9 VEHICLES' is displayed.

implémenté des messages de succès et d'erreur dans les formulaires et après les formulaires.



Base de données

La base de donnée présente 2 tables à ce jour, une pour les véhicules et l'autre pour les utilisateurs (mais aucun service n'a encore été fait pour celle là).

Notez que les champs ont été conçus et générés en suivant les automatismes et bonnes pratiques de Symfony.

Voici notre modèle d'entités-relations :

Vehicle		User	
PK	<code>id int NOT NULL</code>	PK	<code>id int NOT NULL</code>
	<code>type char(64) NOT NULL</code>		<code>email char(180) UNIQUE NOT NULL</code>
	<code>stock int NOT NULL</code>		<code>roles json NOT NULL</code>
	<code>characsJSON char(255) NOT NULL</code>		<code>password char(255) NOT NULL</code>
	<code>imageName char(255)</code>		
	<code>updatedAt DateTime NOT NULL</code>	Pas encore de relations	

Difficultés :

Une des difficultés de ce projet est l'apprentissage de Symfony. Ce n'est pas un framework facile à prendre en main, son fonctionnement est assez opaque, mais nous nous accrochons. Ce qui fut le plus difficile fut de trouver des guides pour l'apprendre qui soient suffisamment adaptés et dont la réalisation s'effectue dans un environnement de développement similaire au nôtre afin de produire une codebase cohérente. En effet, nous avons remarqué que certaines dépendances du projet (mysql, php) sont très sensibles à de tels changements, et que des erreurs

cryptiques sont vites arrivées. Heureusement, dans la présente version de notre projet nous n'avons pas détecté de bogues.

Choses à faire :

Une étape qu'on doit ajouter est le login / logout. C'est essentiel mais nous n'y avons pas encore touché, afin d'éviter de faire bugger le site. Nous ne pouvons pas encore nous authentifier, donc tout le monde peut modifier les annonces du loueur, ce qui est un peu contraignant. C'est une chose sur laquelle nous essayons de travailler afin de l'avoir pour la version finale de notre site.

Compétences développées :

Plusieurs compétences ont été développées grâce à ce projet. Tout d'abord, la prise en main et l'apprentissage de Symfony. Nous avons suivi plusieurs vidéos (dont celles d'une chaîne youtube appelée "Les Teachers du Net", qui a été très utile et que nous recommandons pour les années suivantes), ce qui nous a permis de suivre les bonnes pratiques avec Symfony. Notre code est bien structuré et suis les pratiques de sécurité et de maintenabilité.

Nous avons aussi utilisé le système de versionning Git et le site Github, afin de nous partager les fichiers et leurs modifications.

Le travail d'équipe a aussi été un enjeu très important car dans un aussi court laps de temps, nous avons dû nous adapter les uns par rapport aux autres pour travailler ensemble et mener à bien une première version de ce site de location de voitures.

Ce qu'il reste à faire :

La structure du code est bonne donc il n'y a pas de refactor de prévu sur le code déjà écrit.

Maintenant nous allons ajouter les fonctionnalités de login et logout que nous savons déjà faire, avec Symfony ce sera rapide. Puis nous devons ajouter les relations loueur/véhicules. Et finalement nous devons ajouter les fonctionnalités de réservation.

Bilan :

Nous avons fait une grande partie du travail, mais il ya encore des fonctionnalités à implémenter

Nous pensons qu'avec le travail déjà effectué et l'apprentissage solide de Symfony que nous avons suivis, nous avons une bonne vision d'ensemble qui nous aidera à finir le projet. Les plus grandes difficultés sont derrière nous.

Annexes :

Page principale

The wireframe illustrates the main page flow:

- Step 1: Vehicle Selection**
 - A sidebar on the left shows a list of vehicles with a placeholder image and the text "Nom caractéristiques".
 - An arrow points from the first item to the text "Clic sur un véhicule sélectionné".
 - The selected vehicle is highlighted with a larger placeholder image and the text "Mon caractéristiques".
- Step 2: Vehicle Overview**
 - The main area displays a grid of vehicle cards under the heading "(Sélectionné à l'étape 1)".
 - Each card contains a placeholder image and the text "Nom caractéristiques".
 - The last card in the grid contains the text "(Sélectionné à l'étape précédente) Nom caractéristiques".
 - A large empty box labeled "plus" is positioned to the right of the grid.
- Step 3: Confirmation**
 - A bottom bar contains the text "Confirmer location".

Browse New vehicle Login Register

Vehicle successfully updated!

9 VEHICLES

Activities Firefox Web Browser Oct 19 12:18 PM 800 vehicle-PT - Mozilla Firefox

Edit vehicle-PT

Stock: 10 Status: Unavailable

Characteristics (JSON)

```
>{"weight":10, "year":2004, "price":100000}
```

Save

Activities Firefox Web Browser Oct 19 12:18 PM 800 vehicle-PT - Mozilla Firefox

Edit vehicle-PT

Stock: 10 Status: Unavailable

Characteristics (JSON)

```
{}  
{"weight":10, "year":2004, "price":100000}
```

Save

Activities Firefox Web Browser Oct 19 12:18 PM 800 vehicle - Mozilla Firefox

New vehicle - Mozilla Firefox

New vehicle

Type:

Image (JPG or PNG file): No file selected. Browse...

Stock: 10 Status: Available

Characteristics (JSON)

```
{}
```

