

RODRIGUES Samuel  
GRICE Samson  
MOLDOVAN Alexandru  
LEGRAND Alexandre

CAR-A-OK

# Projet Développement Web

30 mai 2021



# Table des matières

<b>1</b>	<b>Architecture du projet</b>	<b>2</b>
1.1	Front-end . . . . .	2
1.2	Back-end . . . . .	2
<b>2</b>	<b>Utilisation du site</b>	<b>3</b>
2.1	Préparation a l'exécution . . . . .	3
2.2	Execution du serveur . . . . .	3
<b>3</b>	<b>Conclusion</b>	<b>3</b>
3.1	Nos avis personnels . . . . .	3
3.1.1	Alexandru . . . . .	3
3.1.2	Samuel . . . . .	3
3.1.3	Samson . . . . .	4
3.1.4	Alexandre . . . . .	4

# 1 Architecture du projet

## 1.1 Front-end

Nous voulions que l'esthétique du site ait un aspect semi-professionnel, tout en restant propre et organisé. Au départ, nous voulions faire tout le CSS à la main afin d'avoir une liberté au niveau de ce que nous voulions montrer. Cependant, nous avons rapidement compris que la réalisation d'un site web réactif n'était pas une tâche facile, et puisque nous étions autorisés à utiliser des frameworks, nous avons décidé d'essayer Bootstrap. Au début, Bootstrap était excellent. Il nous a permis d'obtenir un site web essentiellement réactif sans avoir à écrire des pages de CSS. Bootstrap fonctionnait parfaitement lorsqu'il était utilisé dans des situations simples tel que dans la création de formulaire ou de tableau. Cependant, ce n'était pas aussi fonctionnel dans des situations plus complexes, comme pour la barre de navigation ou les images coulissantes de la page principale. Pour ces situations, nous étions obligés de revenir au CSS classique : c'est-à-dire rédigé à la main.

Comme le front-end a besoin d'un back-end opérationnel pour pouvoir créer correctement les pages, nous avons commencé à travailler sur certains des éléments dont nous savions que nous aurions besoin, comme la page principale, la barre de navigation, le pied de page et beaucoup de formulaires. On est parti sur un thème vert sachant que la couleur invoque une atmosphère paisible. Afin de souligner cette atmosphère vis-à-vis de l'utilisateur, nous avons nommé le site 'CAR-A-OK' avec l'idée en tête que le mélange d'un fond vert et l'imagerie de la musique est un effet rassurant. De même, la conception du logo repose sur le nom du site et sur le type assurance que le site représente : c'est-à-dire la musique et les voitures. Pour ce qui est des formulaires mentionnées précédemment, nous les avons créés à l'aide de Bootstrap, ce qui a permis d'obtenir des formulaires facilement modifiables et réactifs. Nous avons créé un certain nombre de formulaires que nous pouvions utiliser comme modèles pour le site, mais il s'est avéré que lorsque nous les avons intégrés au site, il y avait tellement de choses à ajuster et à modifier que nous aurions pu tout aussi bien ne pas nous embêter avec les modèles. D'un autre côté, les modèles que nous avons créés pour la barre de navigation et le pied de page ont été extrêmement efficaces puisque nous pouvions les inclure dans chaque page et qu'il suffisait de modifier le modèle pour effectuer une mise à jour à l'échelle du système. Considérant le contexte d'une compagnie d'assurance, il est important que les personnes impliquées soient averties le plus tôt que possible quand quelque chose se passe. Nous avons ainsi envisagé de faire un système de notification même si ce n'était pas nécessaire. La cloche dans la barre de navigation permet d'ouvrir une fenêtre modale affichant les messages non lus en vert foncé et les messages lus en blanc. L'utilisateur a également le choix de marquer toutes les notifications comme lues et le choix de toutes les supprimer. Sinon pour chaque notification, l'utilisateur doit simplement cliquer sur la notification pour être redirigé vers la messagerie.

## 1.2 Back-end

Nous n'avions pas le droit aux bases de données ni à d'autres frameworks PHP. Nous avons donc décidé de créer nos propres classes pour la gestion des données, à base de fichiers json. Nous avons adopté une approche visant à sécuriser les données en ayant le contrôle total dessus. Nous avons donc décidé de créer un module Router, permettant de créer nos propres URLs et rediriger le trafic comme bon nous semble. Cela implique donc de changer le comportement de base de PHP, qui cherche à associer les requêtes à un système de fichier. Pour que ce routage soit possible, nous avons donc besoin de rediriger toutes les requêtes à travers un fichier index, qui lui va ensuite s'occuper du reste. Nous voulions également simplifier l'affichage des éléments basiques tels la barre de navigation, le footer ou d'autres parties, afin de faire cela, nous avons décomposé le site en plusieurs fichiers partiels : nous pouvons ainsi par exemple directement inclure la navbar en haut de chaque page. Le traitement des données envoyées et reçues étant primordial dans la sécurité d'un site web, chaque requête et type de donnée est vérifié grâce à un système de permissions, de classes et autres fonctions de traitement. Les fichiers mis en ligne par les utilisateurs sont par exemple renommés, filtrés et ne sont jamais exécutés. Afin de permettre la bonne communication entre le frontend et le backend, toutes les requêtes sont au format JSON et un effort a été fait pour standardiser les formats. Un fichier de configuration est disponible afin d'obtenir un site web le plus modulaire possible, permettant le changement des chemins des fichiers par défaut, le niveau de logs à garder, ainsi que d'autres paramètres de personnalisation.

Pour ce qui est de la génération des QRCode nous avons décidé d'utiliser une bibliothèque tierce développée par [chillerlan](#). Cela nous a permis de ne pas trop nous préoccuper de la logique derrière la création d'un code QR et ainsi de pouvoir poursuivre le développement d'autres fonctionnalités.

## 2 Utilisation du site

### 2.1 Préparation a l'exécution

Avant de lancé le serveur il faut d'abord s'assurer qu'un certain nombre de fichier sont créé dans la base de donné. Nous avons intentionnellement choisi de ne pas créer ces fichier à l'exécution pour s'assurer qu'aucun fichier potentiellement dangereux puissent être créé par un utilisateur malveillant. Ainsi il faudra crée les fichier et dossier suivants si il n'existe pas déjà (Si le projet a était cloner depuis notre github par exemple). Sous linux nous avons créer un script bash pour pouvoir préparer et exécuté le serveur comme il faut.

- data/
  - uploads/
  - tickets/
  - messages/
  - assurances.json
  - contracts.json
  - conversations.json
  - tickets.json
  - sell.json
  - sinistres.json
  - users.json
- logs/

FIGURE 1 – Listes des fichiers et dossiers a préparer

Il faudra ensuite configuré l'adresse du serveur dans le fichier config/settings.php a l'entrée 'url' du premier define afin de s'assurer que les QRcodes redirigent bien vers la bonne adresse.

### 2.2 Execution du serveur

Étant donné le routage que nous avons implémenter dans notre projet notre site doit être lancé depuis un seul fichier en utilisant PHP-CLI. la commande a exécuter est spécifié dans le README qui vient avec le code du projet.

## 3 Conclusion

### 3.1 Nos avis personnels

#### 3.1.1 Alexandru

Avec Samuel, nous ne sommes occupés de créer l'infrastructure du projet et de rendre celle-ci robuste mais modulaire. L'approche du router et l'utilisation des classes m'a permis de développer mes connaissances en PHP et de découvrir de nouvelles notions. J'ai ainsi pu redécouvrir et travailler avec les headers, les MIME types, la gestion des exceptions, les scopes des classes et plus généralement les "features" du PHP. Je suis par ailleurs toujours à la recherche du fichier php.ini sur ma machine. Je suis globalement content du rendu final, entre le routeur fonctionnel prenant en charge les expressions régulières, le système de partials permettant des horreurs telles l'insertion de php dans des fichiers js et les trop nombreuses pour être listées fonctionnalités de la messagerie. Le temps donné pour la réalisation de ce projet n'était pas vraiment assez important pour la taille de ce projet, cependant nous avons réussi à le compléter avec des bonus et atteindre plus de 600 commits. Finalement, j'ai apprécié les nombreuses heures passées sur ce projet et l'expérience a été plutôt positive.

#### 3.1.2 Samuel

Comme Alexandru l'a dit nous nous somme principalement occupés le l'infrastructure pour le rendre modulaire. Mon travail était donc de rendre la base de donné facile d'accès et modulaire afin de pouvoir rajouter un type de donné facilement et pouvoir y accéder de manière efficace et sécurisé sans compromettre la

lisibilité du code. Pour cela j'ai grandement utilisé les Classes qui sont très utiles pour créer des structure de données de manière simple et consistante. En effet on peut grâce a elle facilement vérifier les données entrées et créer des interfaces pour les modifier de manière sur. De plus j'ai aussi beaucoup appris sur la structure d'un serveur web en général en effet avec notre système de routeur nous avons du nous documenté sur le routage HTTP ainsi que trouvé des solution à nos problématique. En voulant rendre notre code stable j'ai aussi appris a me servir des structures try catch en PHP permettant de traiter des erreur lors de l'exécution et ainsi réagir comme il faut a celle-ci.

Globalement je me rend compte que ce projet était un vrais défis en particulier car nous avions de grande ambitions. Nous avons malgré tout pu arrivé à un point ou nous somme satisfait de ce que nous avons produit, Et malgré les nombreuses difficulté rencontré lors de notre réalisation et le manque de temps, je suis content de ce que j'ai pu apprendre et faire sur ce projet.

### 3.1.3 Samson

Mon rôle dans le projet était de rendre le site esthétique avec Alexandre. Je me sentais assez à l'aise pour faire le CSS du site et j'étais impatient de voir comment je pouvais adapter mes compétences pour inclure Bootstrap également. Comme le front-end devait attendre que le back-end soit opérationnel avant que je puisse commencer à travailler sur le CSS réel, j'ai commencé à travailler sur certains modèles de base dont je savais que nous aurions besoin. Par exemple, j'ai créé certains formulaires, comme le login, la modification des informations de l'utilisateur et la possibilité de créer un utilisateur. Cependant, lorsqu'ils ont été mis en place sur le site, ils ont tellement changé parce qu'ils nécessitaient des entrées par ajax ou qu'il leur manquait des informations, qu'il n'y avait malheureusement pas besoin de ces modèles. J'ai aussi commencé à aider Alexandre avec le css pour la barre de navigation pendant que je travaillais sur les images coulissantes de la page principale et le pied de page. Une fois le backend opérationnel, j'ai pu commencer à créer les pages utilisateur du site, ce qui impliquait de comprendre comment utiliser l'API qu'Alexandru et Samuel avaient développée. Cela signifiait beaucoup d'AJAX pour récupérer l'information correcte du serveur et aussi pour envoyer l'information à la requête correcte.

J'ai apprécié ce projet final pour dev web, même si j'ai trouvé que la quantité de travail était bien trop importante, même en un mois, avec de nombreux autres projets qui se terminent et les examens finaux qui arrivent. Néanmoins, nous avons bien travaillé ensemble et avons réussi à créer quelque chose qui n'est pas trop mal, si je peux me permettre.

### 3.1.4 Alexandre

Au départ, faire partie de ce groupe était un privilège. Alexandru m'avait fait la proposition de travailler avec lui en début d'année au projet final qu'il avait anticipé. Mon recrutement se basait sur son appréciation de la valeur esthétique que je pouvais apporter au site web et également mes compétences en développement web. Au départ, cela me posait problème puisque cela signifiait plus ou moins trahir mon groupe habituel. Mais après avoir appris qu'on pouvait faire des groupes interclasses, sachant que Samuel et Samson faisaient également partie du groupe, j'ai vu cette proposition en tant qu'opportunité de sortir de ma zone de confort et d'être entouré des meilleurs programmeurs de la PRÉPA 2. J'ai la chance de pouvoir m'améliorer et d'obtenir de l'expérience dans une atmosphère plus professionnelle. Alexandru et Samuel m'avaient donné des conseils par rapport aux front-ends disant que je devrais utiliser le framework appelé 'Bootstrap Studio'. J'ai essayé de m'initier à ce programme mais je n'avais pas acquis les bases en CSS pour me sentir avec le framework. Ainsi, j'ai suivi le conseil de Samson et j'ai solidifié mes bases en CSS en le faisant à la main. De cette manière, j'ai pu fournir la barre de navigation et le bas de page qu'on utilisera fréquemment. Or au fil des semaines, malgré les entretiens réguliers que j'avais avec Samson (qui s'occupait également de la partie front-end du site), je ne voyais pas comment je pouvais apporter du mien étant un peu déconnecté du groupe que j'étais le seul membre de l'équipe en distanciel. Cela était aussi dû au fait que je ne savais pas comment utiliser GitHub. La seule initiative que j'ai prise était avec la marque du site tel que le nom, les slogans, le logo et les palettes de couleurs. Alexandru m'avait tâché de concevoir la fenêtre des notifications pour la messagerie. Ce que j'ai fourni était accepté avec appréciation par mes collègues mais ma contribution au site, comparé à mes camarades, me paraissait minime, ce qui est quelque chose auquel je n'y étais pas habitué. Néanmoins, les progrès que j'ai fait en CSS étaient visibles grâce au site que j'ai fait pour la réalisation de mon TIPE.