

Dossier de projet titre développeur web et web mobile

K-line auto

ZIDI Mohamed



**Développement d'un site pour faciliter la prise de rendez-vous et
d'une interface administrateur**

Sommaire :

Introduction	04
1.Présentation	05
1.1 Présentation personnelle	05
1.2 Présentation La Plateforme_ Coding School.	05
1.3 Présentation de la formation	05
2.Présentation du projet	06
2.1 Présentation de l'entreprise	06
2.2 Compétence couvertes par le projet	07
2.3 Résumé du projet	08
2.4 Cahier des charges	08
2.4.1 Les objectifs du site	08
2.4.2 Les fonctionnalités du site	08
2.4.3 Les cibles	09
2.4.4 Le périmètre du projet	10
2.5 Arborescence du site	10
3. Développement de la partie frontend de l'application	11
3.1 Maquette	11
3.2 Charte graphique	11
3.3 Développer une l'interface utilisateur web dynamique	12

3.3.1 Intégration et contenu dynamique	13
3.3.2 Responsive design	14
3.4 Conclusion	15
4. Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité	16
4.1 Organisation	16
4.2 Conception de la base de données	17
4.3 Développement de la partie BackEnd d'une application mobile/ web mobile	19
4.3.1 Connexion base de données et héritage	19
4.3.2 Fonctionnalité significative back office	20
4.3.3 Fonctionnalité significative prise de rendez-vous	22
5.Veille sur les vulnérabilité de sécurité	23
5.1 Référencement des principales failles de sécurité existantes	24
5.2 Pratiques suivies pour sécuriser le site	26
5.3 Conclusion	30
6.Recherches anglophone	31
7.Annexes	31

Introduction

En septembre 2021, j'ai intégré une formation dans le but de préparer mon passage au titre professionnel (RNCP de niveau 5) de développeur web et web mobile, à LaPlateforme_ Marseille.

Avec l'objectif de me présenter à ce titre professionnel, de proposer un support de lecture complet et d'acquérir les compétences professionnelles requises pour, j'ai travaillé sur le développement d'un site permettant la prise de rendez-vous pour une entreprise de lavage automobile pour particuliers et professionnels.

Le projet que je vous présente dans ce dossier a été pour moi une première expérience dans le monde professionnel dans le développement web et web mobile. En effet, devoir élaborer un site web complet en commençant par le maquettage, la rédaction du cahier des charges, ensuite passer à la conception de la base de données et au développement. Le tout en gardant le lien et l'échange avec l'entreprise pour laquelle je réalise ce projet pour à terme satisfaire le client.

1. Présentation

1.1 Présentation personnelle

Je me nomme ZIDI Mohamed et je suis actuellement en reconversion professionnelle suite à un contrat de cinq années dans l'armée de terre. J'ai choisi de m'orienter vers ce milieu car je souhaitais reprendre des études dans un domaine qui attise ma curiosité et mon envie d'apprendre.

1.2 Présentation La Plateforme_ Coding school

La Coding School est une formation web qui s'adresse à tous ceux qui souhaitent s'ouvrir les portes des métiers du numérique. Le modèle pédagogique unique de l'École la Plateforme, s'adapte aux besoins de chacun. Les évaluations se font par des contrôles continus sur des projets webs concrets réalisés seuls ou en groupes. La coding school revendique une pédagogie active et inductive centrée sur l'apprenant et orientée projet.

1.3 Présentation de la formation

La formation (1200 h/an) s'effectue en présentiel, dans un lieu dédié spécifiquement pour catalyser l'apprentissage au 8 rue d'Hozier 13002 Marseille. Le programme vise à dispenser les connaissances et compétences nécessaires pour acquérir l'obtention du titre professionnel : technologies du web, maquettage

d'applications, modélisation de bases de données, développement de sites web statiques, dynamiques et responsives, déploiement de CMS, base d'algorithmie, projet professionnel.

2. Présentation du projet

2.1 Présentation de l'entreprise

K-line Auto est une entreprise de lavage automobile située au 97 Bd Rouvier dans le 10ème arrondissement de Marseille, fondée en 2021. Elle promeut un lavage intérieur et extérieur de tout type de voiture à la main avec un savoir-faire qui lui est propre.

Pour faciliter et centraliser la prise de rendez-vous ainsi qu'augmenter sa présence sur internet, la création d'un site web dédié était devenue la suite logique des choses.

2.2 Compétences couvertes par le projet

Ci-dessous, vous trouverez les compétences nécessaires à la validation du titre de développeur web et web mobile :

Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

- Maquetter une application
- Réaliser une interface utilisateur web statique et adaptable
- Développer une interface utilisateur web dynamique
- Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce

Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité

- Créer une base de données
- Développer les composants d'accès aux données
- Développer la partie back-end d'une application web ou web mobile
- Elaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce

2.3 Résumé du projet

La naissance du projet s'est faite suite à des échanges avec le directeur de K-line Auto concernant les problématiques qu'il souhaitait régler à l'aide d'un site internet. Tout d'abord, il souhaitait un site permettant de faciliter la prise de

rendez-vous pour le client ainsi et de centraliser celle-ci car jusqu'ici elle était divisée en plusieurs procédés tels que la prise de contact via les réseaux sociaux ou bien la réservation sur groupon.

Ces échanges ont également mené au désir de l'équipe K-line Auto de posséder un espace administrateur ergonomique et simple d'utilisation sur leur futur site permettant de gérer la suppression et l'ajout de contenu et de rendez-vous.

2.4 Cahier des charges

Il est nécessaire de rédiger un cahier des charges assez conséquent pour s'assurer du bon déroulement d'un projet d'une telle envergure.

2.4.1 Objectifs du site

Le site a pour objectif de faire augmenter le chiffre d'affaires de l'entreprise et le nombre de clients potentiels. Le site aura aussi pour objectif de limiter la prise de contact sur des plateformes autres que le site internet.

2.4.2 Les fonctionnalités du site

Le site comportera une partie **vitrine**, une partie **réservation** et une partie **administrateur**.

La partie **vitrine** sera composée de la page d'accueil, une page qui présente les différents services proposés par K-line Auto ainsi qu'une galerie photos.

La partie **réservation** comportera un formulaire de prise de rendez-vous avec plusieurs inputs sécurisés: les informations du client, son type de véhicule, la formule choisie, la date et l'heure. Une fois le formulaire rempli et envoyé, le client ainsi que l'administrateur recevront un mail de confirmation avec toutes les informations nécessaires.

La partie **administrateur** contiendra une liste des rendez-vous, la possibilité d'ajouter, de modifier ou de supprimer les services proposés, les types de véhicules concernés et de déplacer ou annuler les réservations.

2.4.3 Les cibles

Les **utilisateurs** du site web seront :

- Les personnes qui possèdent une voiture qui ont connu K-line Auto via les réseaux sociaux ou le bouche à oreille.
- L'équipe K-line Auto.
- Les professionnels souhaitant contacter K-line Auto pour un partenariat ou un contrat pour plusieurs véhicules.

L'application doit être facile à utiliser pour tout type d'**utilisateurs**.

2.4.4 Le périmètre du projet

Le site sera réalisé en français et ce dernier devra être accessible sur différents supports, à savoir mobile, tablette et ordinateur.

2.5 Arborescence du site

L'arborescence du site se décline comme ceci :

- Page d'accueil
- Page présentant les services
- Page galerie photo
- Page contact
- Page prise de rendez vous

Une partie back office est également prévue afin de permettre la gestion du site, elle aura son arborescence propre:

- Page inscription administrateur
- Page connexion administrateur
- Page tableau de commande

3. Développement de la partie frontend de l'application

Avant de commencer le développement front de mon projet il fallait tout d'abord choisir une charte graphique et réaliser une maquette, et pour cela j'ai effectué quelques recherches sur google pour m'inspirer du design de certains sites déjà existants. Je mettrai des liens vers les sites que j'ai sélectionné dans les annexes du dossier.

3.1 Maquette

La maquette a été réalisée avec le logiciel gratuit Figma.

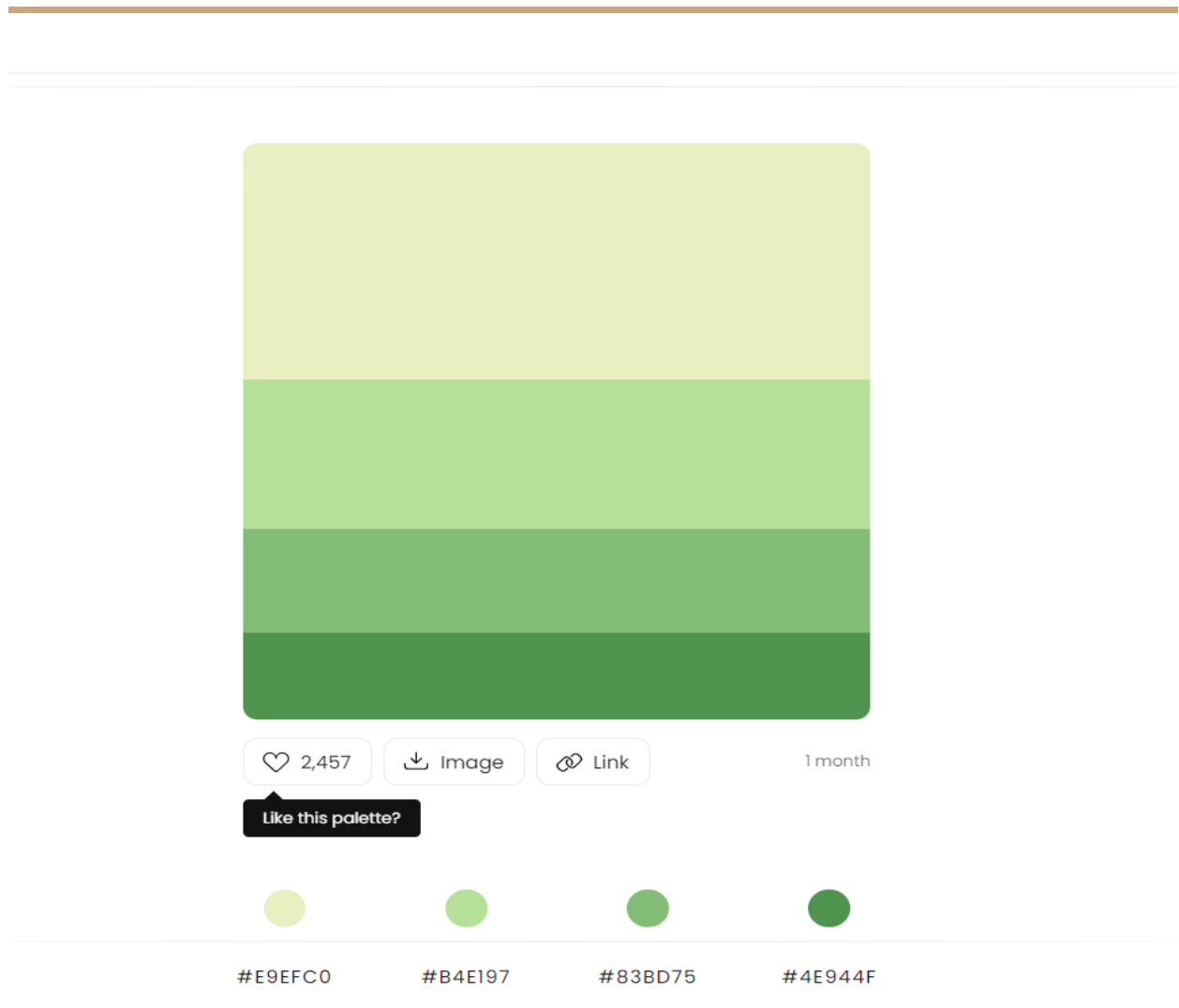
L'entreprise n'ayant pas de Web designer, j'ai réalisé moi même le design du site. Le gérant m'a laissé carte blanche pour réaliser ce site, je lui ai quand même envoyé des captures d'écran de la maquette pour qu'il la valide. Je me suis inspiré du design de différents sites d'entretien automobile déjà existants pour faire cette maquette. J'ai également créé une version mobile de cette maquette.

Vous trouverez des images de la maquette dans les annexes du dossier.

3.2 Charte graphique

La police d'écriture est la suivante: Lato.

La palette de couleurs dominante est la suivante:



J'ai choisi cette palette sur colorHunt car je désirais faire un design orienté autour du vert et lumineux.

3.3 Développer une interface utilisateur web dynamique

J'ai utilisé **HTML/CSS** comme structure des pages à développer et pour l'intégration web. Le style des pages est réalisé en **CSS**. Pour que le design des pages s'adapte aux différents supports (mobile/desktop), on peut parler de design responsive, j'ai

utilisé les **media queries**. Le contenu des pages est généré ou s'adapte à l'utilisateur grâce au langage PHP.

3.3.1 Intégration et contenus dynamiques

Index : C'est une page web **statique**. Cette page a été pensée comme une introduction au site. Elle y présentera sans entrer dans les détails les différentes caractéristiques ainsi que les services proposés par l'entreprise. Des **liens href** vers les autres pages du site et surtout la prise de rendez vous sont disposés.

Header : Il est commun à toutes les pages du site auxquelles le client a accès. Il est divisé en 2 balises **div** contenues dans une nav, la première contient le logo et la deuxième les différents liens pour accéder aux autres pages et aux réseaux sociaux de l'entreprise. Sur les plus petits écrans les liens sont accessibles via un **bouton burger**.

Footer : Le footer est composé de deux colonnes, une partie navigation qui permet de naviguer sur le site ainsi qu'une partie réseaux sociaux. Il est aussi commun à toutes les pages auxquelles le client a accès.

Nos services : Elle met en avant les différents services proposés par K-line Auto, ils sont aussi détaillés pour que le client puisse se faire une idée sur la formule qu'il choisira. Il y a aussi plusieurs liens **href** dirigeant vers la page prise de rendez-vous.

Réserver un lavage : Page contenant un formulaire avec divers **inputs** permettant au client d'entrer ses informations personnelles, de choisir une formule de lavage et une date ainsi qu'une heure pour son rendez-vous.

Validation de réservation : Une fois que le client a rempli correctement le formulaire de la page réserver un lavage, il aura accès à cette page qui adapte les

informations qu'elle affiche en fonction de l'utilisateur. Elle synthétise toutes les informations importantes du client telles que la date et l'heure de rendez-vous, son numéro de réservation.

Page contact : un formulaire de contact qui sera récupéré par l'admin qui contient toutes les entrées de l'utilisateur dans ce formulaire. Ces informations sont affichées grâce à des méthodes de la **classe admin**.

Page connexion-admin : ce sont des formulaires avec **les inputs** indispensables à l'authentification. Des échanges avec la bdd sont effectués lors de la validation du formulaire permettant **d'afficher des messages d'erreur** en cas d'entrée d'informations erronées.

Page admin: Une fois connecté, l'admin pourra injecter, supprimer et modifier du contenu. Il pourra accéder à tous les rendez-vous. Cette page possède son propre design et une navigation propre à elle.

Galerie : Page qui contient diverses photos d'avant et après par exemple. Permet aux clients de se faire un avis sur les résultats promis par l'entreprise et de mettre en avant tout le savoir-faire K-line Auto.

3.3.2 Responsive design

Afin de faciliter l'**adaptation de l'application web aux tablettes et mobiles**, j'ai utilisé des **display flex** pour composer les pages et l'ensemble de ses éléments. Je définis également la taille des éléments en pourcentage pour que celle-ci s'adapte le plus possible.

J'ai utilisé l'inspecteur de Chrome afin de pouvoir visualiser l'application sur les différents formats d'écran. En fonction de la taille de l'écran j'ai modifié les dimensions et la disposition des différents éléments de chaque page avec les **Media Queries**.

J'ai changé le type de **mise en page en fonction de la taille de l'écran**. Au lieu d'avoir une seule mise en page pour toutes les tailles d'écran, la mise en page est modifiée. Les éléments sont repositionnés, les typographies réduites, les images redimensionnées pour les écrans plus petits.

3.4 Conclusion

Une fois que la charte graphique et la maquette ont été validées par mon client, j'ai procédé à l'intégration web et web-mobile. J'ai réalisé le développement de la partie front en desktop-first, or de plus en plus de personnes naviguent sur internet uniquement à l'aide de leur téléphone, pour de prochains projets je pense qu'il serait plus judicieux de commencer par un développement orienté mobile-first.

4. Développement de la partie back-end de l'application web ou web mobile en intégrant les recommandations de sécurité

4.1 Organisation

Pour développer la partie back-end, j'ai choisi d'utiliser la technologie **PHP** et plus précisément la programmation orienté objet(**P.O.O**).

J'ai programmé la partie connexion admin, inscription admin, le back office, la gestion et l'utilisation de PHPmailer. J'ai réalisé le formulaire qui permet d'envoyer les rendez-vous en base de données et directement dans l'espace administrateur.

J'ai également géré l'envoi de mail via le formulaire de la page contact qui permet aux clients de poser une question sur les services proposés ou de se renseigner sur l'entreprise et les façons de faire de K-line auto.

Pour m'organiser, j'ai défini une liste de fonctionnalités. Après répartition de ces dernières un tableau de bord sur **Trello** m'a permis de suivre l'avancée de chacune.

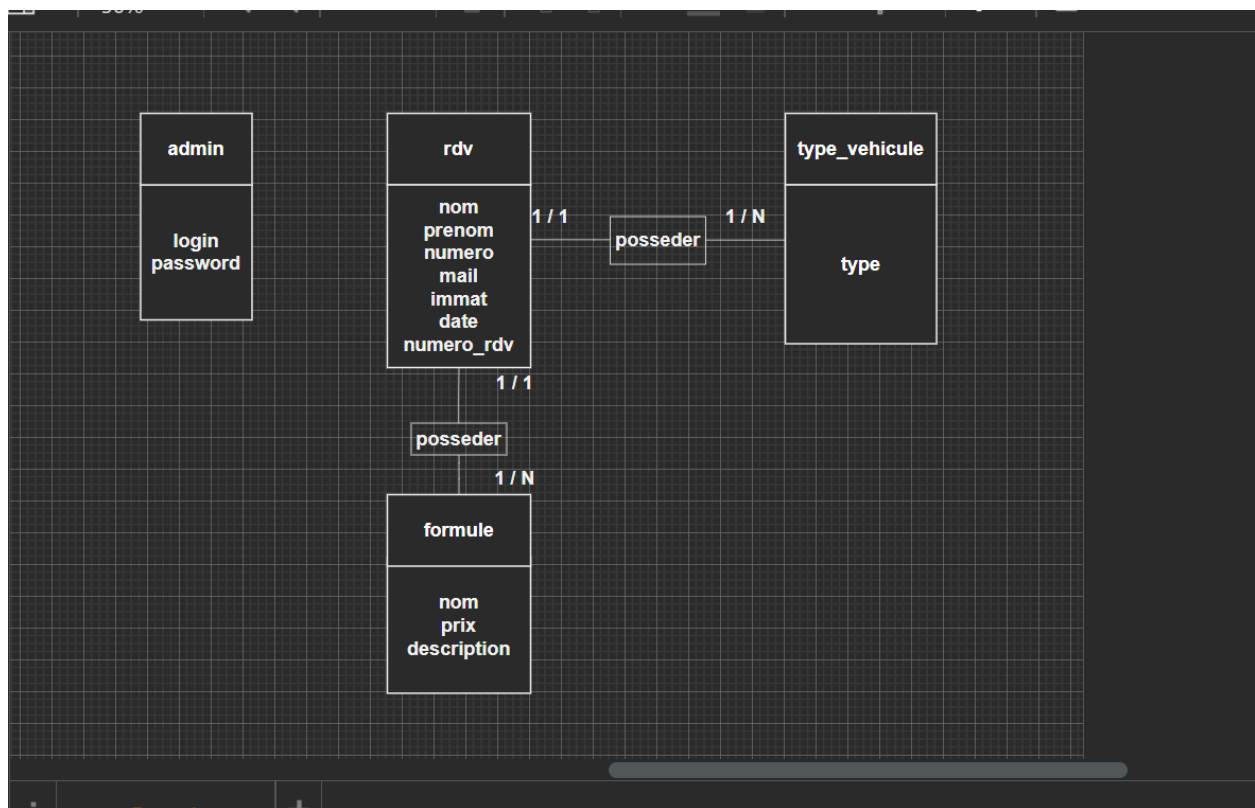
Les outils suivants ont été utilisés pour les différentes étapes du développement de la partie back-end du projet :

- Trello pour le suivi des tâches

- Visual Studio Code pour l'IDE
- L'extension drawio de VS Code pour la conception de la base de données
- phpMyAdmin pour la création et la gestion de la base de données
- Wamp pour la mise en place d'un serveur en local

4.2 Conception de la base de données

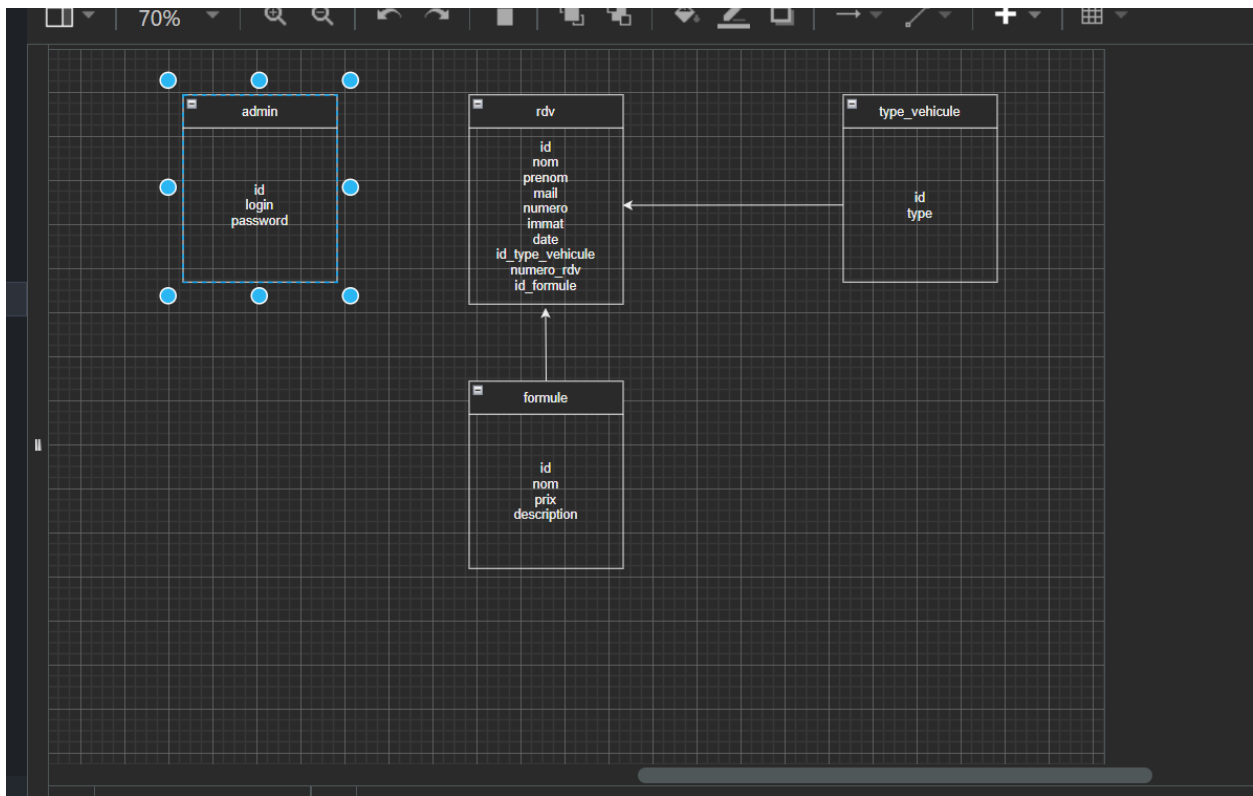
Au regard des fonctionnalités demandées par l'entreprise, j'ai développé la base de données suivante :



Modèle conceptuel de données

Comme illustré ci-dessus sur le **modèle conceptuel de données**, on peut voir que la base de données s'articule autour de quatre **entités principales** reliées

entre elles par des **associations**. Des **cardinalités** existent également entre ces entités.



Modèle logique de données

Après avoir conçu le modèle conceptuel il faut le convertir en **modèle logique de données** avant de commencer le développement **back end**. Pour cela, les entités deviennent des **tables**, les **cardinalités** disparaissent, les **clés étrangères** apparaissent dans les tables et les **associations** laissent place à des flèches montrant quelles tables donnent leur **clé primaire** à une autre, celles qui possédaient les cardinalités 0/n ou 1/n à l'étape précédente de la **conception**.

C'est avec ces 4 tables que je gère les données pour par exemple enregistrer les rendez-vous pour ensuite les afficher dans le panel admin avec la table **rdv**,

permettre aux administrateurs de gérer les informations sur les formules proposées à l'aide de la table **formule** et de gérer l'inscription et la connexion d'administrateurs grâce à une table dédiée.

4.3 Développement de la partie back end d'une application mobile/ web mobile

4.3.1 Connexion base de données et héritage

Concernant la connexion à la base de données j'ai procédé de la manière suivante.

J'ai codé une class **Bdd** qui se connecte à la base de données dans `_CONSTRUCT` directement, j'ai utilisé un **"try and catch"**, la fonction php qui gère les erreurs. La gestion d'une erreur via une exception se fait en deux temps.

On va utiliser un bloc try dans lequel le code qui peut potentiellement retourner une **erreur** va être exécuté. On crée à l'intérieur une nouvelle connexion grâce à l'objet **new PDO**.

On va créer un bloc catch dont le but va être d'attraper l'exception si celle-ci a été lancée et de définir la façon dont doit être gérée l'erreur. La fonction est appelée dans les autres classes afin de réaliser différentes requêtes.

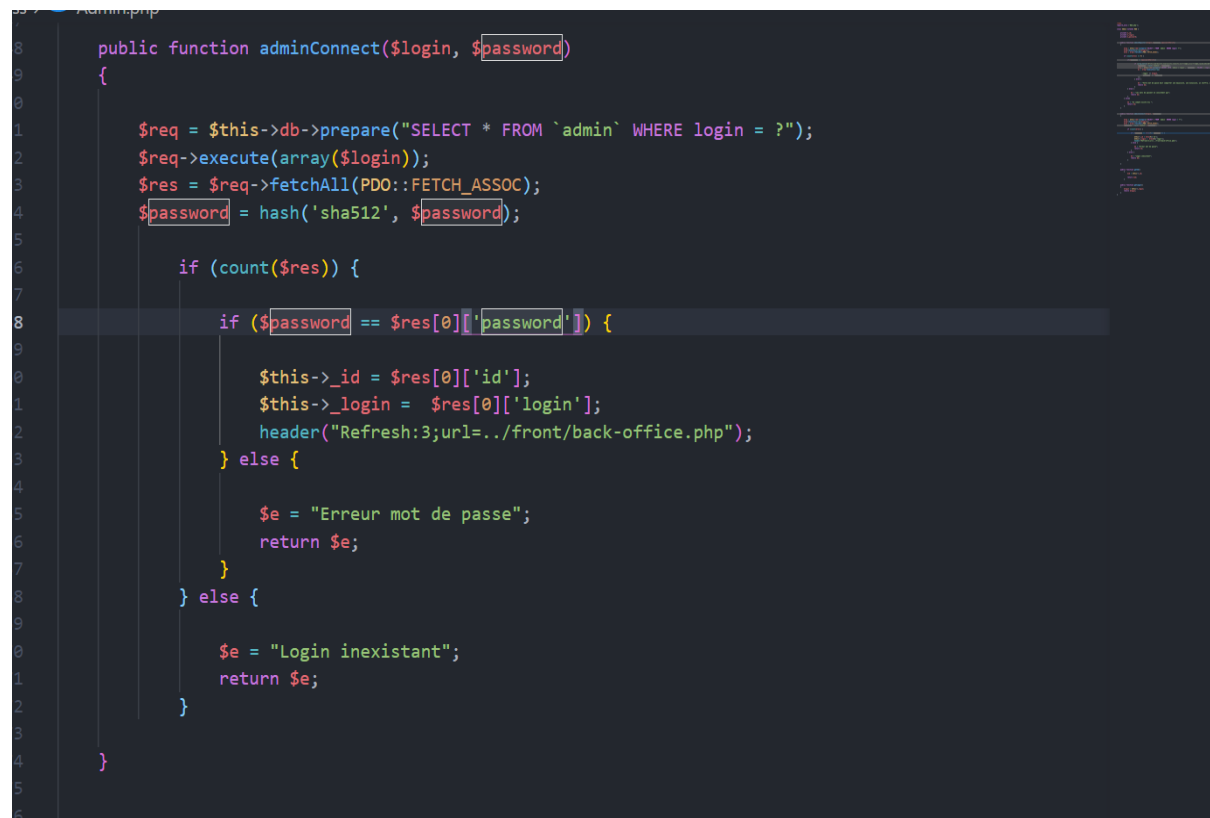
Grâce au principe d'héritage de la programmation orientée objet de PHP, je fais hériter cette class Bdd à toutes mes autres classes qui ont besoin d'une connexion à la base de données pour ne pas répéter mon code, et pouvoir réaliser différentes requêtes.

4.3.2 Fonctionnalité significative back office

La page admin est l'une des artères principales du site internet, en effet beaucoup d'opérations non visibles par un utilisateur lambda seront visibles par l'administrateur :

Pour **identifier un utilisateur en tant qu'administrateur** il faut faire appel à la base de données comme une connexion classique. Il s'agit d'interroger la base de données pour savoir si le login et le mot de passe entrés dans les champs du formulaires correspondent à ceux inscrits en base de données. S'ils correspondent, une connexion est établie et une ou des variables de sessions sont initialisées contenant les informations de l'admin connecté que je récupère à l'aide d'une méthode getter.

Extrait de code illustrant une connexion administrateur:

A screenshot of a code editor with a dark background and light-colored text. The code is in PHP and defines a function named 'adminConnect'. The function takes two parameters: '\$login' and '\$password'. It starts by preparing a SQL query to select all columns from an 'admin' table where the login matches the provided '\$login'. It then executes this query and fetches all results as an associative array. Next, it hashes the provided '\$password' using 'sha512'. The code then checks if the number of results from the query is greater than zero. If so, it compares the hashed password with the password stored in the database for the given login. If they match, it sets session variables for '_id' and '_login' and sends a 'Refresh' header. If the passwords don't match, it sets an error message '\$e' to 'Erreur mot de passe' and returns it. If the login doesn't exist, it sets '\$e' to 'Login inexistant' and returns it. The function ends with a closing brace for the function definition.

```
7  
8 public function adminConnect($login, $password)  
9 {  
10  
11     $req = $this->db->prepare("SELECT * FROM `admin` WHERE login = ?");  
12     $req->execute(array($login));  
13     $res = $req->fetchAll(PDO::FETCH_ASSOC);  
14     $password = hash('sha512', $password);  
15  
16     if (count($res)) {  
17  
18         if ($password == $res[0]['password']) {  
19  
20             $this->_id = $res[0]['id'];  
21             $this->_login = $res[0]['login'];  
22             header("Refresh:3;url=../front/back-office.php");  
23         } else {  
24  
25             $e = "Erreur mot de passe";  
26             return $e;  
27         }  
28     } else {  
29  
30         $e = "Login inexistant";  
31         return $e;  
32     }  
33 }  
34  
35 }  
36
```

Bien évidemment, avant de pouvoir se connecter et accéder à son espace, l'administrateur devra s'inscrire. Pour cela il devra choisir un **login** et un **mot de passe**. Si le login choisi est déjà pris, l'inscription sera impossible. J'ai également utilisé une expression régulière (**REGEX**) pour qu'il soit obligé de choisir un mot de passe de minimum 8 caractères contenant une majuscule, une minuscule, un chiffre et un caractère spécial. Une fois inscrit, il sera redirigé vers la page connexion administrateur. Bien entendu cette url ne sera transmise et accessible qu'au gérant de K-line auto.

Extrait de code illustrant une inscription administrateur:

```
1  <?php
2  require_once ('Bdd.php');
3
4  class Admin extends Bdd {
5
6      private $_id;
7      private $_login;
8      private $_password;
9
10
11     public function adminRegister($login,$password,$passwordVerify)
12     {
13
14         $req = $this->db->prepare("SELECT * FROM `admin` WHERE login= ?");
15         $req->execute(array($login));
16         $res = $req->fetchAll(PDO::FETCH_ASSOC);
17
18         if (count($res) == 0) {
19
20             if($password == $passwordVerify){
21
22                 if (preg_match('#^(?=.*[A-Z])(?=.*[a-z])(?=.*\d)(?=.*[~!*$@%_])([~!*$@%_\w]{8,35})!');
23                 $password = hash('sha512', $password);
24                 $req = $this->db->prepare("INSERT INTO `admin`(`login`, `password`) VALUES (:log
25                 $i = $req->execute(array(
26
27                     ':login' => $login,
28                     ':password' => $password
29                 ));
30                 header('location:../front/admin-connect.php');
```

4.3.3 Fonctionnalité significative prise de rendez-vous

J'ai, parmi toutes mes **classes**, une qui gère tout ce qui est en lien avec la table **rdv** de ma base de données. Elle me permet entre autres d'insérer en base de données toutes les informations des réservations faites par les différents clients, de récupérer toutes les réservations pour les afficher au niveau du **back office**.

J'ai utilisé deux expressions régulières pour vérifier que le numéro de téléphone ainsi que la plaque d'immatriculation du client soient conformes. Si un rendez-vous est déjà réservé à la même heure, le code ne s'effectuera pas et l'utilisateur verra un message d'erreur pour lui indiquer que cette réservation est impossible.

Une fois le rendez-vous inséré en base de données, j'ai fait en sorte à l'aide de **PHPmailer** qu'un mail contenant toutes les informations importantes soit envoyé au client et au gérant de K-line auto directement sur leur boîte mail respective.

Extrait de code illustrant une inscription administrateur:

```
public function insertRdv($nom,$prenom,$mail,$tel,$immat,$type_vehicule,$formule,$date){

    $req = $this->db->prepare("SELECT * FROM `Rdv` WHERE date = :date");
    $req->execute(array(
        ':date' => $date
    ));
    $res = $req->fetchAll(PDO::FETCH_ASSOC);

    if(count($res)==0) {

        if (filter_var($mail, FILTER_VALIDATE_EMAIL)){

            if(preg_match('#^0[6-7]{1}[0-9]{8}$#',$tel)){

                if(preg_match('#^[A-Z]{2}[-][0-9]{3}[-][A-Z]{2}$#',$immat) || preg_match('#^[0-9]{3}[-][A-Z]{2}$#',$immat)){

                    $numero_rdv = rand(0,1000000);
                    $request = $this->db->prepare("INSERT INTO `rdv`(`nom`,`prenom`,`numero`,`mail`,`id_type_vehicule`,`numero_rdv`,`id_formule`,`date`,`immatriculation`)");
                    $exec = $request->execute(array(
                        ':nom' => $nom,
                        ':prenom' => $prenom,
                        ':numero' => $tel,
                        ':mail' => $mail,
                        ':id_type_vehicule' => $type_vehicule,
                        ':numero_rdv' => $numero_rdv,
                        ':id_formule' => $formule,
                        ':date' => $date,
                        ':immatriculation' => $immat
                    ));
                }
            }
        }
    }
}
```

5. Veille sur les vulnérabilités de sécurité

Il est important de travailler sur la veille sur la sécurité pour ce projet. Je me suis appuyé sur la riche documentation de l'OWASP et ai effectué différentes recherches pour lister les principales failles et trouver les solutions pour les contrer.

5.1 Référencement des principales vulnérabilités

Oubli de valider les entrées des utilisateurs. Injections dans les formulaires.

La plus basique et sûrement la plus connue est l'injection SQL. Deux tiers des attaques sur le web portent dessus.

- **Contrôle d'accès inefficace.**

Il s'agit de la possibilité pour un pirate d'utiliser l'identité d'autres personnes sur un site Internet, si des éléments permettant d'authentifier un utilisateur (login / mot de passe) sont mal protégés dans le site web. La deuxième possibilité est de pouvoir deviner ou modifier les éléments d'authentification facilement.

- **Mauvaise gestion des sessions.**

Cette faille exploite des faiblesses dans les mécanismes qui permettent au serveur du site web de se souvenir de qui vous êtes, une fois que vous vous êtes authentifié.

- **Cross Site Scripting**

Cette faille touche les sites web qui laissent les internautes publier du code HTML susceptible d'être vu par les autres utilisateurs du site (dans un forum, par

exemple). Cela permet d'exécuter des contenus dynamiques sur les navigateurs, avec les droits associés au site web.

- **Dépassement de mémoire tampon ou buffer overflow**

Une faille qui consiste en une corruption de la mémoire, bien souvent la mémoire de la pile des appels. La plupart du temps, le programme va planter, mais ceci ouvre aussi une porte au hacker qui veut contrôler un processus à distance.

- **Injection de commandes**

L'injection de commandes(ou Shell Code Injection) est une attaque qui consiste à exécuter des commandes systèmes non autorisées sur le système d'exploitation d'une victime via une application vulnérable.

- **Désérialisation non sécurisée (Insecure Deserialisation)**

Une vulnérabilité de type "insecure sérialisation" permet à un utilisateur malveillant d'accéder et de modifier les fonctionnalités de l'application ciblée.

Mauvaise utilisation du chiffrement

Pour stocker des informations sensibles il faut les convertir en une chaîne de caractères illisible, on dit haché, pour qu'elles ne soient plus lisibles de manière irréversible.

- **Utiliser un logiciel ou des composants présentant des vulnérabilités**

Lorsqu'une faille est découverte, les développeurs de l'application en question proposent généralement un patch qui permet de corriger le problème. Cependant, si la mise à jour n'est pas faite, l'application s'expose à la faille.

- **Défaut dans la configuration des paramètres de sécurité**

Elle est due à une configuration par défaut non sécurisée, des configurations incomplètes, des messages d'erreurs contenant des informations sensibles.

5.2 Pratiques suivies pour sécuriser le site

Les normes consultées :

- ISO/IEC 27000
- RGPD règlement général sur la protection des données
- L'Open Web Application Security Project (OWASP) est un organisme impartial, mondial et sans but lucratif. Il évalue les dix principaux risques pour la sécurité des applications web et préconise un développement logiciel sécurisé.

Sécurisation de l'application contre l'injection SQL

- Les requêtes préparées : on peut écrire les requêtes SQL en paramétrant les variables. C'est ce qu'on appelle une requête préparée. Préparer la requête permet de l'exécuter une fois que l'on a stocké les arguments à envoyer en base de données. Une vérification sur le type de données que l'utilisateur a entré est ainsi effectuée en amont.

exemple de requête préparée dans la class admin pour un INSERT en bdd :

```
$req = $this->db->prepare("INSERT INTO `admin`(`login`, `password`) VALUES (:log:");
$i = $req->execute(array(
    ':login' => $login,
    ':password' => $password
));
```

Protection des données stockées sur l'application :

Les algorithmes de hachage pour crypter les données : Il existe de nombreux algorithmes de hachage : Bcrypt, Scrypt, SHA, MD5, Argon5 et PBKDF2, par exemple. Crypter le mot de passe avec le hachage permet de générer une empreinte unique pour une entrée. Cependant, cela n'empêchera pas le phishing qui reste une méthode très utilisée par les hackers pour récupérer les mots de passe des utilisateurs.

Exemple de hachage de mot de passe dans la classe admin pour l'inscription d'un administrateur

```
$password = hash('sha512', $password);
```

- **import de fichiers** : La faille upload est un risque rencontré lorsqu'on permet à un utilisateur de télécharger des documents sur le site web. Il est notamment aisé de télécharger un document contenant un malware sur le site web ou la base de données. L'administrateur pouvant importer des photos, quelques tests sur le fichier à télécharger (format, taille etc) ont été effectués. Cependant, il ne faut pas se fier uniquement au nom du fichier, car il est possible de nommer un fichier d'une façon trompeuse, telle que "fichier.exe.jpg" pour passer le filtre avec succès.

Empêchez le piratage de session

- demande d'un mot de passe fort pour les utilisateurs contenant des majuscules, des minuscules, des chiffres et des caractères spéciaux
- accès limité permettant l'accès à certaines parties du site uniquement aux personnes ayant les droits suffisants. Pour chaque page, des vérifications sont faites : l'utilisateur est-il connecté, est-il un administrateur, etc. Ces vérifications permettent ainsi de limiter le risque

que des personnes non habilitées puissent accéder à certaines informations sensibles.

Autres préconisations à effectuer sur les versions suivantes de l'application :

- demande aux utilisateurs qu'ils changent régulièrement leur mot de passe en cas d'attaque de credential stuffing ;
- implémentation d'une authentification forte, c'est-à-dire avec plusieurs facteurs d'authentification, comme la validation par SMS ou par mail, par exemple.
- Le cas des cookies et des sessions :
 - s'assurer que les cookies sont chiffrés lors de la transmission via HTTPS ;
 - pas de stockage d'informations d'identification en texte clair dans les cookies
 - définir une date d'expiration pour les cookies-session.
 - ne pas mettre pas l'ID de session dans l'URL ;

-
- PHP possède une bibliothèque appelée SessionManager avec des fonctions qui peuvent être utilisées pour valider les sessions avec des restrictions
 - Le certificat SSL pour protéger le mot de passe lorsqu'il est transmis sur le réseau. Obtenir un certificat SSL et l'ajouter au serveur. Ce certificat est nécessaire pour chiffrer les données en cours de transmission.
 - Sécurisation avec l'API OWASP

L'organisation OWASP dispose d'une API appelée the OWASP Enterprise Security API (ESAPI). Elle peut être utilisée pour sécuriser vos applications web

- Utilisation d'un pare-feu d'application web ou WAF, pour Web Application Firewall. Ce pare-feu se place entre l'utilisateur et l'application web et permet de vérifier et d'intercepter les données envoyées.

5.3 Conclusion

La sécurité est un enjeu majeur dans le développement d'une application web. Il est indispensable de mettre en place toutes les stratégies possibles à la sécurisation des données dans toutes les étapes du code. Cependant, d'autres implémentations,

citées précédemment, sont nécessaires pour améliorer la sécurité du site pour sa mise en ligne.

6. Recherches anglophone

Pour gérer le sélecteur de date(datepicker) sur la page prise de rendez-vous j'ai utilisé la librairie jquery et jquery UI. N'ayant jamais utilisé ces librairies pour faire de genre de choses auparavant, j'ai effectué sur google des recherches en anglais de préférence, car les réponses sont plus nombreuses.

J'ai effectué les recherches suivantes :

"datepicker jquery"

"datepicker jquery stackoverflow"

Ce qui m'a permis de trouver la documentation sur le datepicker jquery et de pouvoir gérer les différents paramètres de celui-ci.

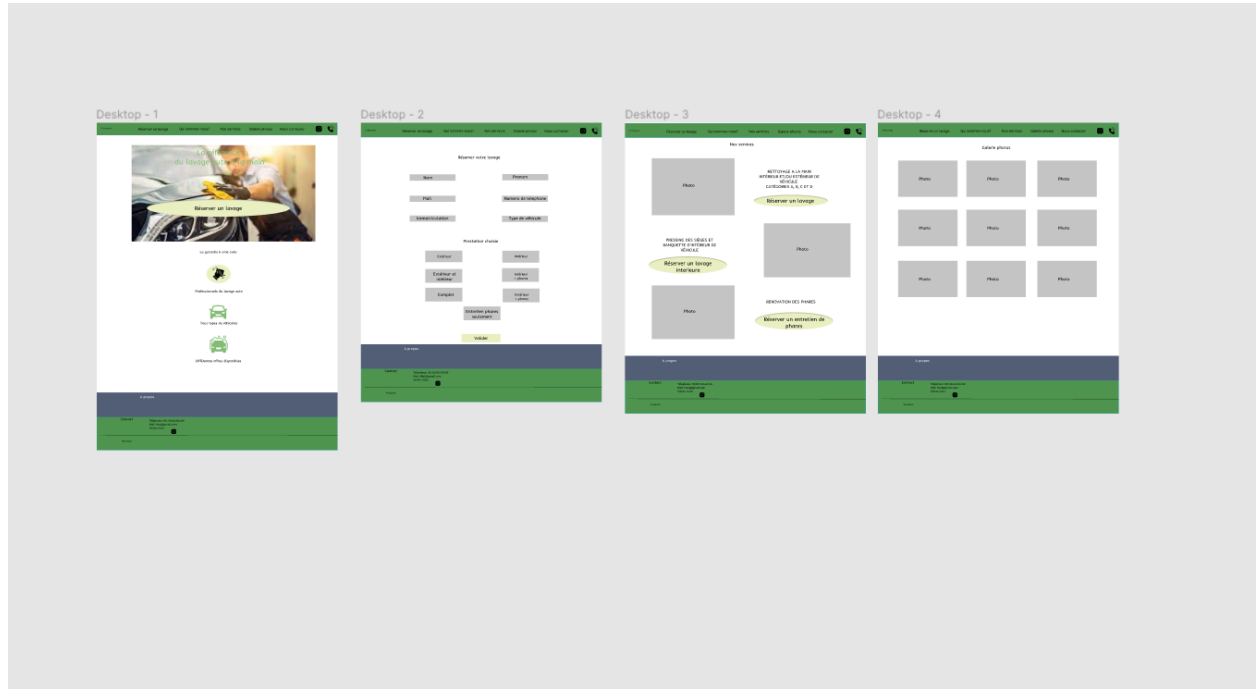
7. Annexes

Liens vers les sites desquels je me suis inspiré pour le design:

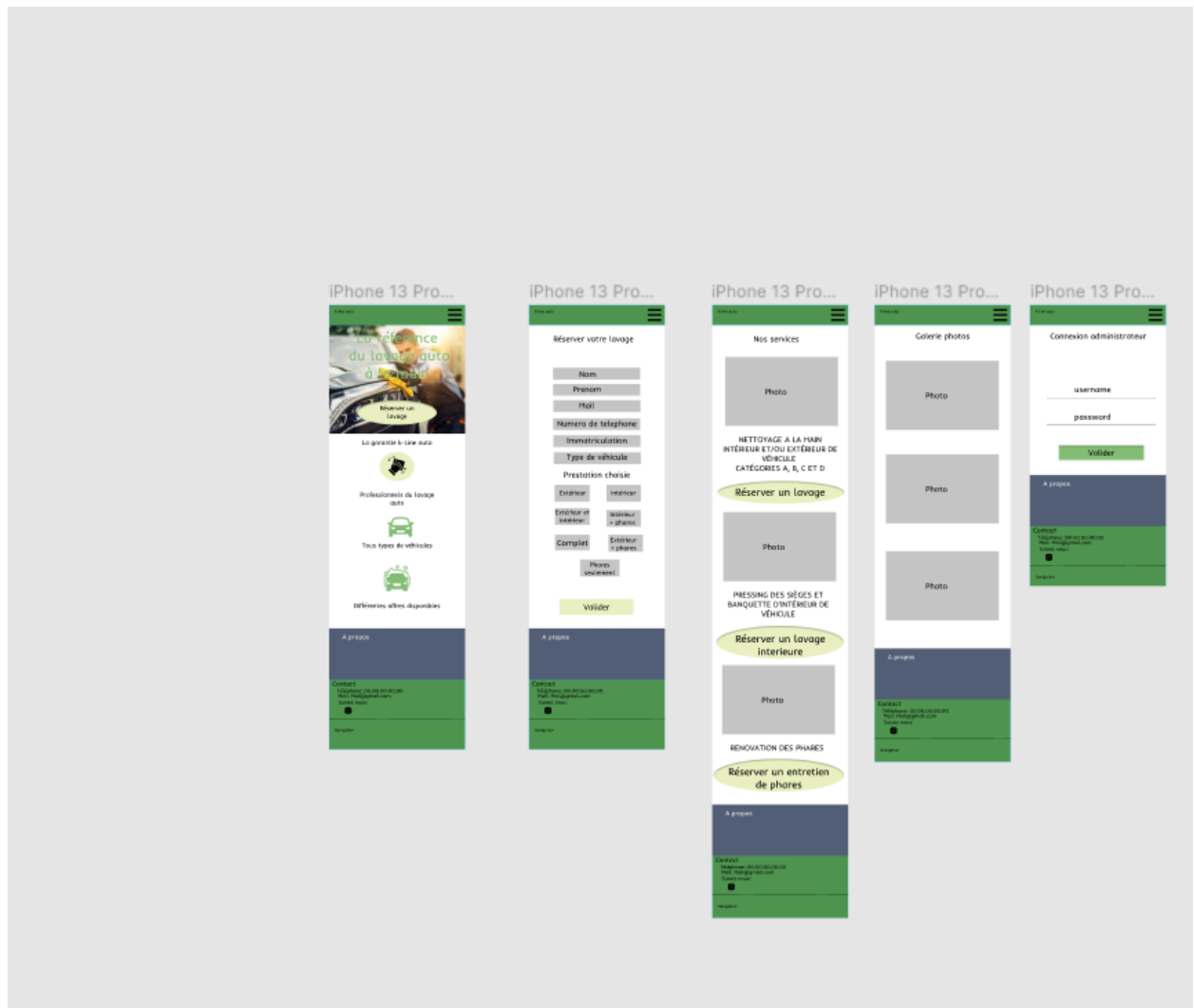
<https://www.washmee.fr/>

<https://www.cosmeticar.fr/>

Extraits de la maquette



Version desktop



Version mobile