



**PROJET CHEF D'ŒUVRE**

**DEVELOPPEUR WEB ET WEB ET WEB MOBILE**

**CERTIFICATION AU TITRE PROFESSIONNEL RNCP DE NIVEAU 3**

**FORMATION DEVELOPPEUR WEB FULL STACK JS**

**SIMPLON.CO SAINT-QUENTIN-EN-YVELINES**

**DU 26/12/18 AU 31/07/19**



# TABLE DES MATIÈRES

Table des matières .....	2
REMERCIEMENTS .....	5
QUI SUIS-JE.....	6
FORMATION .....	7
1. PRÉSENTATION DE SIMPLON.CO.....	7
2. CANDIDATURE ET SÉLECTION A LA FORMATION.....	7
3. DÉBUT DE LA FORMATION.....	7
4. LA PÉDAGOGIE SIMPLON.CO.....	8
STAGE.....	9
PRÉSENTATION DU PROJET .....	11
PROJECT SUMMARY .....	12
.....	12
LISTE DES COMPÉTENCES DU RÉFÉRENTIEL.....	13
1. FRONT-END.....	13
1.1 Maquetter une application .....	13
1.2 Réaliser une interface web statique et adaptable .....	13
1.3 Développer une interface utilisateur web dynamique .....	13
1.4 Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce.....	13
2. BACK-END .....	14
2.5 Créer une base de données .....	14
2.6 Développer les composants d'accès aux données .....	14
2.7 Développer la partie back-end d'une application web ou web mobile .....	14

2.8 Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce .....	14
CAHIER DES CHARGES .....	15
1. CONTEXTE.....	15
2. OBJECTIF.....	15
3. CIBLE .....	15
4. LES OBJECTIFS QUANTITATIFS .....	15
5. PÉRIMÈTRE .....	16
6. FONCTIONNEMENT .....	16
7. CONTENU .....	16
8. CONTRAINTES TECHNIQUES.....	17
9. DÉLAI .....	17
10. USE CASE.....	17
11. MAQUETTES.....	18
12. CHARTE GRAPHIQUE.....	18
MÉTHODES AGILES .....	19
OUTILS DE DÉVELOPPEMENT .....	20
1. VISUAL STUDIO CODE.....	20
2. GIT / GITHUB / GIT BASH .....	20
3. WORKBENCH .....	20
4. GIMP .....	20
ENVIRONNEMENT DE DÉVELOPPEMENT.....	21
1. BASE DE DONNÉES .....	21
2. FRONT-END.....	23
EJS.....	23
JAVASCRIPT .....	24

BOOTSTRAP .....	25
JQUERY .....	27
CSS .....	28
CDN .....	29
RÉFÉRENCEMENT.....	30
RGPD .....	31
3. BACK-END .....	32
NODE.JS .....	32
EXPRESS.JS .....	32
NODE MODULES .....	32
ARCHITECTURE MVC .....	33
FONCTIONNALITÉ REPRÉSENTATIVE .....	34
SITUATION AYANT NÉCESSITÉ UNE RECHERCHE .....	39
TRADUCTION .....	40
VEILLE SUR LES VULNÉRABILITÉS DE SÉCURITÉ .....	41
ANNEXES .....	42
USE CASE.....	42
ARBORESCENCE DES PAGES WEB DU SITE PROJECT CARS.....	43
MAQUETTES DE LA LANDING PAGE VERSION DESKTOP .....	44
MAQUETTE DE LA LANDING PAGE VERSION PC (SUITE) .....	45
MAQUETTE DE LA LANDING PAGE VERSION TABLETTE.....	46
MAQUETTE DE LA LANDING PAGE VERSION MOBILE.....	46
MODELE CONCEPTUEL DE DONNEES .....	47

# REMERCIEMENTS

Je tenais tout d'abord à remercier Mme **Aurélia Perrot** cheffe de projet réseau des fabriques Ile-de-France chez Simplon.co qui m'a aidé dans mes démarches d'inscription et de présélection à cette formation.

Un grand merci à **Sara Guerric**, **Sébastien Besnier** et **Brian Boudrioux**, nos formateurs, pour leurs qualités techniques et pédagogiques. Et d'avoir su aussi nous remotiver lors des moments difficiles.

**Rami Mahfoudh** mon mentor de chez Adsearch pour ces précieux conseils dans ma recherche de stage.

**Lila Benhammou** directrice générale de **Humans4Help** et ma tutrice de stage **Nawel Hebbar** pour la confiance qu'elles m'ont accordées durant mon stage dans leur start-up.

Et enfin je remercie **Simplon.co** et le **conseil départemental des Yvelines** de proposer ces formations gratuitement à un public éloigné de la sphère du numérique.

## QUI SUIS-JE

Je me nomme Wicem Smondel 37 ans, je suis en situation de reconversion professionnelle. J'ai quitté les bancs de l'école à 18 ans après avoir échoué au BAC informatique de gestion dans un lycée classé ZEP de Vitry-sur-Seine dans le Val-de-Marne.

J'ai enchaîné les emplois réclamant peu de compétences (manutentionnaire, déménageur, agent de sécurité, livreur...), mais j'avais toujours en moi l'envie de reprendre mes études, car j'avais le sentiment de n'avoir pas pu exprimer pleinement mes capacités. Durant un séjour en Guadeloupe j'en ai profité pour passer un **BAC marketing** en candidat libre que j'ai obtenu avec mention à l'âge de 25 ans, j'ai par la suite tenté d'intégrer un BTS en alternance commerce international, bien qu'ayant réussi les tests d'entrée de 2 différentes écoles je n'ai jamais trouvé d'entreprise partenaire.

### **Mes expériences professionnelles les plus notables sont les suivantes :**

**Associé gérant** d'un complexe touristique en Tunisie composé de chambres d'hôtes, d'un café maure, d'une crêperie et d'un garage/lavage automobile. Durant 3 ans j'ai géré et managé une dizaine d'employés.

**Associé salarié** d'une société de rénovation de boiseries et sols que j'ai développé dans la ville de Nice après mon retour de Tunisie.

Ces 2 expériences m'ont données envie d'apprendre à gérer une PME par moi-même, j'ai donc repris mes études et obtenu en 2017 un **BTS gestion de la PME** à l'âge de 35 ans. J'ai souhaité poursuivre sur ma lancée avec une licence professionnelle dans le numérique, mais celles-ci n'étaient ouvertes qu'aux étudiants de moins de 26 ans. C'est au hasard d'une lecture que je suis tombé sur les formations de développeur web proposées par Simplon.co et ce projet de reconversion m'a immédiatement emballé.

# FORMATION

## **1. PRÉSENTATION DE SIMPLON.CO**

Simplon.co est un réseau de fabriques numériques et inclusives en France et à l'étranger qui a formé plus de 4 500 apprenants aux métiers du numérique depuis 2013.

C'est une entreprise sociale et solidaire qui souhaite faire du numérique un véritable levier d'inclusion et révéler des talents parmi des publics peu représentés dans ce secteur.

Son fondateur Frédéric Bardeau s'est vu remettre le prix de l'entrepreneur social 2018.

## **2. CANDIDATURE ET SÉLECTION A LA FORMATION**

Pour postuler aux formations de Simplon.co, il faut passer une cinquantaine de badges sur le site Codecademy.com et inclure le lien de son profil au questionnaire en ligne de présélection.

Après avoir été retenu à cette présélection, j'ai été convoqué pour passer des entretiens techniques, de motivations, administratifs et un exercice de groupe qui avait pour but d'évaluer nos capacités à travailler en équipe.

## **3. DÉBUT DE LA FORMATION**

Cette formation a débuté fin décembre 2018 dans les locaux fraîchement rénovés du Campus numérique SQY à Élancourt.

Nous sommes la première promotion **de développeurs web** de ce campus que nous partageons avec une autre promotion de **data artisans**. Chacune de ces promotions est composée d'une vingtaine d'apprenants.

#### 4. LA PÉDAGOGIE SIMPLON.CO

La méthode Simplon.co, c'est **une pédagogie active** où l'apprenant est confronté à des mises en situation concrètes via des projets individuels et en groupe. Le collectif est un élément central de l'apprentissage, car l'apprenant est aidé et évalué par ses pairs.

Les formateurs introduisent les nouvelles technologies abordées par un cours suivi d'un **live code** et donnent des cas pratiques à résoudre en **autonomie** ou en **pair-programming** avant d'aider les apprenants en procédant au débogage.



*Façade du nouveau campus numérique de Saint-Quentin-en-Yvelines*





# STAGE

Au bout de 3 mois de formation, nous devions partir en stage durant 1 mois afin d'aiguiser nos compétences, mais surtout pour se familiariser avec le monde de l'IT en entreprise.

Pour trouver ce stage, j'ai développé mon réseau de contacts sur LinkedIn dès le début de ma formation, en ciblant d'autres développeurs web ainsi que les recruteurs IT, jusqu'à atteindre près de 1500 abonnés. Cette stratégie s'est avérée fructueuse, car en un simple post où j'indiquais être à la recherche d'un stage celui-ci a été vu 1600 fois et au bout de 2 jours j'avais déjà un entretien de recrutement dans une start-up de la Défense "**Humans4Help**" spécialisée dans la RPA (Robotic Process Automation), elle est composée de 50 collaborateurs sur Paris, mais dispose aussi de filiales au Canada, États-Unis, Angleterre, Maroc, Sénégal, Singapour. En une année d'existence, elle a réalisé un chiffre d'affaires de plus 10 millions d'euros et ne cesse de croître. Cette ESN n'avait pas pour spécialité le Web, toutefois, elle avait besoin de développeurs pour créer le site de son organisation.

Pour cette mission j'étais accompagné d'un autre développeur web sortant d'une école d'ingénieur, lui aussi en stage, mais pour une plus longue durée. Lorsque j'ai commencé ce stage, seule la page d'accueil du site [humans4help.com](https://humans4help.com) avait déjà été développée. Mon collègue m'a fait une introduction à **Wordpress** durant ma première matinée de travail. Je n'avais jamais utilisé de **CMS** auparavant, mais dès l'après-midi même, je commençais à développer mes premières pages en utilisant le plug-in **Elementor** qui un constructeur de pages qui facilite et accélère grandement le développement. Chaque matin nous faisons un petit debrief en binôme pour nous fixer les objectifs de la journée entraînant une certaine émulation, et chaque semaine la tutrice de stage qui est la responsable du service Web marketing nous faisait part des objectifs à atteindre.

J'ai pris pour habitude de faire des rapports quotidiens sur l'avancée du projet par mail et par la messagerie interne Teams de Microsoft, à notre tutrice ainsi qu'à la directrice générale de l'entreprise. Ma communication et mon implication ont été grandement appréciées, au point que l'on me confie un autre projet en solo. La directrice générale voulant ajouter comme activité à sa société, le recrutement de ressources pour l'IT. Elle désirait avoir une visibilité sur le web, le plus rapidement possible. J'ai eu libre choix des technologies à utiliser, j'avais pour seules ressources des plaquettes détaillant ce projet. Vu la courte deadline et s'agissant que d'un site vitrine ayant pour seule fonctionnalité qu'un simple formulaire de contact, je me suis donc naturellement tourné vers Wordpress. J'ai développé ce site en 3 jours ainsi que sa traduction, mon choix de nom de domaine [talents4it.com](https://talents4it.com) a été retenu ainsi que le logo que j'ai créé alors que je n'ai aucune compétence dans le graphisme. J'ai donc pu retourner rapidement sur le projet que mon collègue continuait de développer en parallèle du mien, car celui-ci était beaucoup plus important.

Durant ce stage, j'ai donc notamment appris à développer sous **Wordpress**, à utiliser le constructeur de pages **Elementor**, le plug-in de SEO **Yoast**, le plug-in de traduction **Polylang**, créer un **thème enfant** pour ne pas perdre le CSS additionnel lors des mises à jour de ce CMS.

Cette expérience s'est avérée très positive, un poste m'a été proposé que j'ai dû refuser, car je voulais finir ma formation et obtenir ma certification avant tout. Je suis toujours en contact avec cette ESN qui est prête à me recruter et qui n'est pas fermée à mon envie de poursuivre ma montée en compétence en alternance pour obtenir le titre professionnel de **Concepteur, développeur d'application**. *(voir évaluation de stage en annexe).*



## PRÉSENTATION DU PROJET

J'ai pour ambition, en plus de ma future carrière de développeur web, de me lancer dans le commerce de véhicules d'occasions avec un associé.

Nous nous spécialiserons dans les véhicules de retour de fin de leasing/crédit-bail que nous achèterons via les plateformes d'offres spécialisées, car ceux-ci sont généralement excellentement bien entretenus et disposent d'un faible kilométrage.

Nous serons présents sur les grands sites de vente automobile (leboncoin, la Centrale...), et nous posterons au sein de chacune de nos annonces le lien renvoyant sur notre site « **Project Cars** » qui est mon projet chef-d'œuvre pour la certification de développeur web et web mobile.

Ce site a pour but de renvoyer une image professionnelle positive de notre activité, et de créer une relation avec nos futurs clients.

Un visiteur pourra **rechercher** et consulter les voitures de son choix dans notre catalogue, s'inscrire à notre newsletter pour être averti des nouveaux arrivages et **créer son compte**.

Un utilisateur inscrit pourra prendre rendez-vous afin de venir voir le véhicule de son choix, cette prise de rendez-vous générera un **mail automatique** de confirmation pour mon associé qui sera chargé de la vente.

Je serai l'administrateur de ce site pour lequel j'ai créé un **back-office** afin de gérer les annonces (**CRUD**), la rédaction et la diffusion de la newsletter et gérer les comptes utilisateurs.



## PROJECT SUMMARY

My ambition, in addition to my future career as a web developer, is to get into the business of second-hand cars with a partner.

We will specialize in return end-of-lease vehicles that we will purchase through specialized offer platforms as these are generally excellently maintained and have low mileage.

We will be present on the major automotive sales sites (leboncoin, la Centrale ...), and we will post in each of our ads the link to our site which is my project masterpiece for certification web and web developer mobile.

This site aims to return a positive professional image of our business, and to create a relationship with our future customers.

A visitor can search and consult the cars in our catalog, subscribe to our newsletter to be notified of new arrivals and create an account.

A registered user can make an appointment to see the vehicle of his choice, this appointment will generate an automatic confirmation email to my partner who will be responsible for the sale.

I will be the administrator of this site for which I created a back office to manage the announcements (CRUD), the writing and the diffusion of the newsletter and to manage the user accounts.



# LISTE DES COMPÉTENCES DU RÉFÉRENTIEL

Ci-dessous la liste des compétences du référentiel de certification validée par mon projet chef-d'œuvre « Project Cars » :

## 1. FRONT-END

### *1.1 Maquetter une application*

J'ai réalisé mes maquettes versions PC, tablette et mobile à l'aide de l'outil **Balsamiq Mockups**.

### *1.2 Réaliser une interface web statique et adaptable*

Les pages statiques de l'interface client ont été développées avec **EJS** et la partie **responsive** est gérée par **Bootstrap**, **flex** et les **media queries**.

### *1.3 Développer une interface utilisateur web dynamique*

Les résultats des recherches sont générés dynamiquement en **HTML** par **JavaScript**

### *1.4 Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce*

J'ai développé une interface client de **back-office** permettant d'insérer, éditer, effacer les annonces via un formulaire.

## **2. BACK-END**

### ***2.5 Créer une base de données***

J'ai utilisé l'interface graphique de **MySQL, Workbench**, pour créer ma base de données relationnelle, générer le diagramme de mon **modèle conceptuel de données** ainsi que le fichier « **dump** » de sauvegarde et restauration de la structure et des données de ma base.

### ***2.6 Développer les composants d'accès aux données***

Mon projet dispose de nombreuses **requêtes SQL** interrogeant ma BDD pour générer dynamiquement le rendu de la partie client et d'autres insérant des données dans ma base.

### ***2.7 Développer la partie back-end d'une application web ou web mobile***

J'ai développé la partie back-end de mon application sous l'environnement **Node.js** en utilisant son framework **Express** pour créer mon **architecture MVC** (modèle, vue, contrôleur).

### ***2.8 Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce***

J'ai développé une interface client de **back-office** permettant d'insérer, éditer, effacer les annonces via un formulaire.

# CAHIER DES CHARGES

## 1. CONTEXTE

La SAS « **Project Cars** » (nom provisoire) sera créée à la rentrée 2019 et aura pour activité principale l'achat/vente de véhicules d'occasions.

Elle sera composée de 2 associés (dont moi-même) avec un capital de départ de 25 000€.

## 2. OBJECTIF

L'objectif de ce site web est de présenter notre entreprise en véhiculant une image professionnelle positive, d'augmenter notre visibilité dans un secteur à forte concurrence et de créer une relation de confiance avec nos futurs clients.

## 3. CIBLE

Nous ciblons pour clientèle, les utilisateurs des plateformes de petites annonces (leboncoin, la Centrale, l'Argus...) ainsi que le réseau de contacts que mon associé et moi avons pu développer le long de nos carrières respectives et nous prospectorons à l'aide de campagne publicitaire.

## 4. LES OBJECTIFS QUANTITATIFS

Nous visons un objectif de trafic de 500 visites par mois pour viser un volume de 40 contacts prenant rendez-vous pour visiter notre showroom, afin d'assurer la pérennité de notre structure en réalisant un chiffre minimum de 4 véhicules vendus par mois.

Le site comportera une dizaine de pages statiques, les autres seront générés dynamiquement selon la sélection de recherche de l'utilisateur.

## **5. PÉRIMÈTRE**

Internet offre un périmètre sans limites, toutefois vu la taille réduite de notre entreprise la majorité de nos ventes auront lieu au niveau régional, la traduction de notre site en langues étrangères n'est d'actualité, mais en cas d'expansion de notre société, nous envisagerons d'y ajouter les langues des pays limitrophes comme l'Allemagne et l'Italie. La Belgique et la Suisse étant majoritairement francophone il sera donc inutile de l'adapter dans leurs autres dialectes régionaux. Le site aura pour nom de domaine une adresse en **.fr** mais vu le faible prix des services d'hébergement en ligne nous achèterons aussi celui en **.com** afin d'éviter les homonymes. Vu que les smartphones sont devenus les terminaux les plus couramment utilisés pour naviguer sur internet, le fait d'adapter ce site web en version responsive est une évidence.

## **6. FONCTIONNEMENT**

Le site devra permettre aux visiteurs de consulter toutes les annonces de nos véhicules en vente ou de faire une recherche selon ses critères et de s'inscrire à notre newsletter, ils devront se créer un compte pour réserver un rendez-vous pour visiter notre parc automobile.

La partie administrateur devra être simple d'accès et intuitive pour que mon associé qui n'a pas de compétences en informatique puisse y interagir.

## **7. CONTENU**

Le site contiendra des icônes provenant du site [fontawesome.com](https://fontawesome.com) et des photos libres de droits du site [unplash.com](https://unplash.com) pour le carrousel d'accueil. Les photos des véhicules mis en vente seront prises par nos soins sur le parc automobile.

La newsletter présentera les nouveaux véhicules entrés en stock et les nouveaux services que nous proposerons au fur et à mesure de notre développement.



## **8. CONTRAINTES TECHNIQUES**

Pour ce site internet, j'ai choisi d'utiliser les solutions suivantes :

- EJS pour gérer mes templates HTML ;
- Le framework BOOTSTRAP développé par Twitter pour le côté responsive de certains éléments, et pour créer des carrousels, formulaires, boutons... ;
- Node.js avec le framework Express pour mon environnement back-end en architecture MVC ;
- MySQL pour la conception et la gestion de ma base de données.

Je serai chargé du déploiement/hébergement de ce site internet d'en assurer la maintenance et le dépannage et aussi de me charger de la formation de mon associé à l'utilisation du back-office en cas d'indisponibilité de ma part.

## **9. DÉLAI**

Nous disposons d'un mois pour développer nos projets, dossier de projet, présentation de projet. C'est afin de respecter cette deadline que je me suis tourné vers les templates EJS au lieu d'utiliser le framework REACT afin de satisfaire à la culture du **delivery** : « *Done is better than perfect !* ».

## **10. USE CASE**

Le cas d'utilisation « use case » présente la manière d'utiliser un **système** correspondant aux **interactions** des différents **acteurs** en vue d'une finalité, il permet de décrire les exigences fonctionnelles du projet

Les acteurs de ce projet sont **le visiteur** dont les droits sont limités à la consultation du site et à l'inscription à la newsletter. **L'utilisateur inscrit** hérite des mêmes droits que le visiteur, mais peut en plus prendre rendez-vous, l'**administrateur** hérite des droits des 2 précédents acteurs et bénéficie des droits d'accès et d'interaction au back-office. Mon associé, **le vendeur** n'agit

pas dans le **système**, mais il reçoit tout de même un mail l'information des rendez-vous que les utilisateurs ont pris. **(voir annexe n°1)**

## 11. MAQUETTES

J'ai conceptualisé l'arborescence du site « Project Cars » à l'aide de l'outil en ligne **gloopmaps** mettant en évidence les rubrique et sous-rubrique du site. J'ai tenté de respecter le critère d'UX design des « 3 clics » permettant à l'utilisateur de trouver ce qu'il cherche en 3 clics. Seule la prise de rendez-vous déroge à cette règle en nécessitant un 4<sup>e</sup> clic selon le chemin qu'il a utilisé pour sa recherche. **(voir annexe n°2)**

J'ai utilisé le logiciel **balsamiq mockup** qui m'a permis de concevoir rapidement mes zoning & wireframes (PC, tablette et mobile) et de construire mes maquettes par-dessus ces wireframes avec le but de créer un design graphique, ergonomique épuré et efficace véhiculant une image de marque crédible pour rassurer et convaincre l'utilisateur. **(voir annexe n°3)**

## 12. CHARTRE GRAPHIQUE

J'ai choisi pour les titres et sous-titres la police d'écriture **POPPINS** et pour les paragraphes **CENTURY GOTHIC**.

Pour les couleurs de l'interface j'ai misé sur le trio gagnant *noir / blanc / gris*, que je trouve élégant et intemporel et pour « casser » ce style noir et blanc j'ai choisi les couleurs primaires *rouge / vert / bleu* pour les boutons.

Toutes ces couleurs sont à leurs valeurs hexadécimales de base pour maximiser le contraste en espérant une meilleure accessibilité aux malvoyants.

NOIR	GRIS	BLANC	ROUGE	VERT	BLEU
#000	#D3D3D3	#FFF	#FF0000	#008000	#0000FF

# MÉTHODES AGILES

Pour la conduite de nos projets respectifs, plutôt que d'utiliser un **Trello** personnel, les formateurs ont choisi de nous faire travailler publiquement en **agilité** pour permettre un contrôle et une amélioration continue de la qualité de nos projets. Le fait que tout le monde puisse voir l'état d'avancée des projets entraîne une émulation de groupe.

La progression d'une étape à l'autre ne se fait pas de manière linéaire, mais **itérative**. Pour chacun de ses « **sprints** » nous disposons d'un Trello grandeur nature avec des post-its pour remplacer les vignettes, divisées en 3 catégories : *TO DO* / *WORK IN PROGRESS* / *DONE*.

Nous introduisons à tour de rôle lors d'un **stand-up** hebdomadaire, quels **objectifs MVP** (minimum viable product) chacun se définissait pour la fin de la semaine et de retranscrire ces tâches sur différents post-its à placer sous la catégorie *TO DO*, chaque jour nous devons finir au moins une tâche définie à passer de la catégorie *WORK IN PROGRESS* à *DONE*. En cas de blocage d'un apprenant, il y avait une 4<sup>e</sup> catégorie *NEED HELP*, ainsi la personne en difficulté pouvait se faire aider soit par un formateur ou un autre apprenant.



Tableau de gestion des projets de la fabrique d'Élancourt

# OUTILS DE DÉVELOPPEMENT

## 1. VISUAL STUDIO CODE

Pour éditeur de code, après avoir utilisé **SUBLIM TEXT** au début de ma formation je suis rapidement passé à **VISUAL STUDIO CODE** de Microsoft qui est un **IDE** (Integrated Development Environment) qui rencontre un vif engouement auprès des développeurs depuis que Microsoft a ouvert une partie de son code (open source), ce qui a permis à sa bibliothèque de modules d'extensions de croître à vitesse grand V, grâce une immense communauté. De plus ce logiciel est régulièrement remis à jour. J'utilise un grand nombre de ces extensions pour accélérer et simplifier mon travail de développement.

## 2. GIT / GITHUB / GIT BASH

Afin de **versionner** mon code et de ne rien en perdre, j'utilise **GIT** pour « **pusher** » mon travail sur un **repository GITHUB** J'ai intégré leur **terminal de commande GIT BASH** au sein de mon IDE VISUAL STUDIO CODE.

## 3. WORKBENCH

**MySQL WORKBENCH** est une **interface graphique** du système de gestion et d'administration de bases de données relationnelles (**SGBDR**) MySQL, qui permet de concevoir ses bases et de générer leurs modèles conceptuels de données. Auparavant nous utilisions **PHPmyADMIN**, j'ai découvert WORKBENCH lors d'une veille et je l'ai introduit au reste de ma promotion au cours une présentation.

## 4. GIMP

GIMP est un outil d'édition et de retouche d'image, je l'ai utilisé pour normaliser les différentes images de mon projet afin de toutes les convertir au format PNG (plus léger), de les rogner et d'éditer leurs résolutions.

# ENVIRONNEMENT DE DÉVELOPPEMENT

## 1. BASE DE DONNÉES

Après avoir réalisé mes use case et maquettes, la création de ma base de données était l'étape à conclure avant de me lancer dans le code.

J'ai utilisé le **système de gestion de base de données relationnelle Oracle MySQL** qui est un des langages de requête structurée SQL (Structured Query Language) le plus répandu et son interface graphique **WORKBENCH** (voir *OUTIL DE DÉVELOPPEMENT*) pour concevoir ma base de données (**database**) composée de **tables** auxquelles j'ai attribué un **préfixe** par souci de sécurité. Pour chacune des colonnes (**column**) j'ai choisi pour **nomenclature** de reprendre le nom de la table sans préfixe suivi d'un underscore ( \_ ) suivi du nom de la colonne.

Chacune de ces tables a pour identifiant (**id**) une clé primaire (**primary key**) ayant un type de données (**datatype**) entier (**integer**) qui est **automatiquement incrémenté** et dont la valeur ne peut être nulle (**not null**)

Pour les clés étrangères (**foreign keys**) permettant d'associer relationnellement les tables entre elles et de créer des requêtes de **jointure**, j'ai choisi pour nomenclature de leurs **index** automatiques afin d'éviter les doublons : fk(pour foreign key)+ underscore ( \_ ) + les 2 premières lettres de la table + un numéro.

L'index permet au moteur de la base de données (ici **InnoDB**), d'accéder plus rapidement à la donnée souhaitée, pour les clés étrangères ces index sont générés automatiquement, mais nous pouvons indexer les colonnes de notre choix notamment celles qui seront régulièrement requêtées.

À la fin de la construction de ma BDD, WORKBENCH permet de générer, le **modèle conceptuel de données** mettant en valeur la structure de la BDD, les relations entre les tables et leurs index. (**voir annexe n°4**)

**WORKBENCH** me permet aussi de tester mes requêtes avant de les intégrer à mon code en vue d'interroger la BDD, de créer une BDD à partir d'un script SQL et enfin de générer et d'exporter un « *DUMP* » en vue de sauvegarder et de restaurer la structure et les données de notre BDD en l'important.

La connexion du serveur à la BDD se fait par ce script du fichier /config/db.js :

```
const mysql = require('mysql');

const connection = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: 'admin',
  database: 'project_cars',
});

console.log('db connected')
module.exports = connection;
```

Script permettant la création de la database `project\_cars` et de la table `t\_booking` :

```
DROP DATABASE IF EXISTS `project_cars`;
CREATE DATABASE `project_cars`;

CREATE TABLE `project_cars.t_booking` (
  `booking_id` int(11) NOT NULL,
  `booking_date` date NOT NULL,
  `booking_time` time NOT NULL,
  `car_id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  PRIMARY KEY (`booking_id`)
) ENGINE = InnoDB;
```



## 2. FRONT-END

### EJS

Pour développer mes **templates** (modèles) **HTML** j'ai utilisé **EJS** pour gérer plus facilement mes **views**. EJS m'autorise à ne pas avoir à recoder les éléments apparaissant toutes les pages du projet, au sein de mon dossier `/views/` j'ai créé un dossier `/partials/` où j'ai mis mon **header** (en-tête) dans un fichier celui-ci inclut le logo du site et la **navbar** (barre de navigation) et un autre fichier pour mon **footer** (pied de page).

Ainsi dans chaque nouveau templates je n'ai plus à recoder ces 2 blocs, je demande simplement les inclure (**include**) en les appelant de cette façon avec les balises `<% %>`:

```
<!-- views/login.ejs -->
<% include partials/header %>

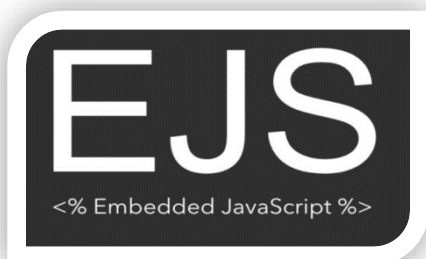
<% include partials/footer %>
```

Les balises `<% %>` permettent d'inclure du code **JavaScript**. Cette ligne de code permet de faire apparaître des messages d'erreurs au sein de ma vue :

```
<% if (typeof error !== 'undefined'){ %><h4 style="color: red"><%=
error%></h4><% } %>
```

Grâce à ces balises j'accéder plus facilement aux *values* d'un formulaire provenant de la base de données :

```
<div class="form-group col-md-4">
  <label for="model">model</label>
  <input type="text" class="form-control" name="model"
id="model" value="<%= t_cars.car_model %>"
```



## JAVASCRIPT

**JavaScript** dans sa version **ES6**, est le langage central de ce projet aussi bien du côté **client** et côté **serveur**.

En front-end j'ai utilisé du JavaScript natif (**Vanilla JS**) pour créer des **constructeurs de classe** afin que les résultats de la recherche effectuée par un utilisateur soient **générés dynamiquement en HTML** :

```
//initialisation - création class Cars et display html
class Cars {
  constructor(model, image, price, energy, doors) {
    this.model = model;
    this.image = image;
    this.price = price;
    this.energy = energy;
    this.doors = doors;
    this.container = document.getElementById("container");
    this.createEl();
    this.setAtt();
    this.appendContent();
    this.insertContent();
    Cars.push(this);
  }

  createEl() {
    this.prodBox = document.createElement('div');
    this.prodmodel = document.createElement('h5');
    this.prodimage = document.createElement('img');
    this.prodDivBottom = document.createElement('div');
    this.prodPrice = document.createElement('span');
    this.buyButton = document.createElement('button');
    this.likeButton = document.createElement('img');
  }

  setAtt() {
    this.prodBox.setAttribute('class', 'cars');
    this.prodmodel.setAttribute('class', 'model');
    this.prodimage.setAttribute('class', 'image');
    this.prodimage.setAttribute('src', this.image);
    this.prodDivBottom.setAttribute('class', 'bottom')
    this.prodPrice.setAttribute('class', 'price');
    this.prodDoors.setAttribute('src', this.doors);
    this.prodEnergy.setAttribute('class', this.energy);
    this.bookingButton.setAttribute('class', 'bookingButton');
  }
}
```



```

appendContent() {
    this.container.appendChild(this.prodBox);
    this.prodBox.appendChild(this.prodmodel);
    this.prodBox.appendChild(this.prodimage);
    this.prodBox.appendChild(this.prodDivBottom)
    this.prodDivBottom.appendChild(this.prodPrice);
    this.prodDivBottom.appendChild(this.buyButton);
    this.prodDivBottom.appendChild(this.likeButton);
}

insertContent() {
    this.prodmodel.innerHTML = this.model;
    this.prodimage.innerHTML = this.image;
    this.prodPrice.innerHTML = this.price + '€';
}

```



## BOOTSTRAP

**BOOTSTRAP** est un **framework CSS** développé par **Twitter**, il permet une meilleure gestion du côté **responsive** réduisant l'usage des **@media queries**.

Il dispose d'une collection d'outils utiles à la création du design qui contient un ensemble de codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs.

BOOTSTRAP découpe la page en **12 colonnes**, il suffit d'annoncer le nombre de colonnes que l'on souhaite utiliser et le format de l'écran (**lg / md / sm / xs**).

Ainsi dans ce **carrousel** j'ai déterminé qu'il prendra toute la largeur de l'écran avec 12 colonnes et que chaque élément n'en prendra que 3 afin d'avoir 4 images les unes à côtes des autres ( $4 \times 3 = 12$ ) :

```

<div class="row">
  <div class="col-md-12 ">
    <div id="desktopCarousel" class="carousel slide" data-
ride="carousel">
      <ol class="carousel-indicators">
        <li data-target="#blogCarousel" data-slide-to="0"
class="active"></li>
        <li data-target="#blogCarousel" data-slide-to="1"></li>
      </ol>
      <!-- Carousel items -->
      <div class="carousel-inner">
        <div class="carousel-item active">
          <div class="row">
            <div class="col-md-3">
              <a href="#">
                
              </a>
            </div>
            <div class="col-md-3">
              <a href="#">
                
              </a>
            </div>
            <div class="col-md-3">
              <a href="#">
                
              </a>
            </div>
            <div class="col-md-3">
              <a href="#">
                
              </a>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```



## JQUERY

**jQuery** est une **bibliothèque** JavaScript (certains diront que c'est un framework) qui raccourcit l'écriture de scripts, de parcourir et modifier le **DOM** plus aisément. Ainsi l'accessibilité aux **classes** et **attributs** et la **gestion d'événement** sont grandement facilitées.

J'ai utilisé jQuery pour le changement de couleur de ma barre de navigation au « Scroll », pour changer la classe de celui-ci en mode responsive et pour modifier la couleur des images des checkboxes de ma barre de recherche au « click ».

```
// Menu-toggle button
$(document).ready(function() {
    $(".menu-icon").on("click", function() {
        $("nav ul").toggleClass("showing");
    });
});

// Scrolling Effect
$(document).on("scroll", function() {
    if ($(document).scrollTop()) {
        $('nav').addClass('black');
    } else {
        $('nav').removeClass('black');
    }
});

//Checkbox color switch
$('.checkBox label img').on('click', function() {
    if ($(this).hasClass('active'))
        $(this).toggleClass('active');
    else
        $(this).next().toggleClass('active');
});
```



## CSS

Le **CSS** est un langage informatique qui décrit la **présentation** des documents HTML, pour en gérer les dispositions, couleurs et surtout les **media queries** qui définissent les techniques d'application du Responsive Web Design en fonction du périphérique, afin d'améliorer l'apparence graphique et la lisibilité.

Media query pour que ma barre de navigation devienne un menu « hamburger » sur tablette et mobile :

```
@media (max-device-width: 1050px) {  
  #logo {  
    position: fixed;  
    height: 90px;  
  }  
  nav ul {  
    max-height: 0px;  
    background: #000;  
  }  
  nav.black ul {  
    background: #000;  
  }  
  .showing {  
    max-height: 100%;  
    z-index: 9;  
  }  
  nav ul li {  
    box-sizing: border-box;  
    width: 100%;  
    padding: 15px;  
    text-align: center;  
    z-index: 9;  
  }  
  .menu-icon {  
    display: block;  
    z-index: 9;  
  }  
}
```



## CDN

Les CDN Content Delivery Network jouent un rôle important dans la vitesse de chargement des pages sur ordinateur ou sur smartphone.

Je préfère utiliser leurs serveurs distants plutôt que de télécharger les fichiers de JQuery, Bootstrap, Fontawesome, les Google Fonts et de les héberger dans le dossier de mon projet et sur mon serveur.

Leur ordre de chargement est important JQuery doit être chargé en premier vu que celui-ci est implémenté dans Bootstrap et la Google font « Poppins » après Bootstrap sinon cette police d'écriture ne fonctionnera pas sur le site.

```
<script src="https://code.jquery.com/jquery-3.4.1.min.js"
    integrity="sha256-
CSXorXvZcTkaix6Yvo6HppcZGetbYMGWSFlBw8HfCJo="
crossorigin="anonymous"></script>
<link
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstr
ap.min.css" rel="stylesheet"
    integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap
.min.js"
    integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
    crossorigin="anonymous"></script>
<link
href="https://fonts.googleapis.com/css?family=Poppins&display=swap"
rel="stylesheet">
<script
src="https://kit.fontawesome.com/2a3b73b180.js"></script>
```

## RÉFÉRENCEMENT

Le **référencement naturel SEO** (search engine optimization) est un ensemble de techniques visant à **optimiser la visibilité** d'une page web dans les résultats des **moteurs de recherche**.

Bien définir la structure de son HTML est un impératif, en utilisant les bonnes balises dans la rédaction de son code, <h1>,< h2>... pour les titres, ne pas oublier la balise <title> pour chacune des pages et les *attributs* « title » et « alt » pour les images.

Il existe ensuite un ensemble de balises <meta> propre au SEO, **viewport**, **description**, **OG**(open graph pour les réseaux sociaux), celle de la **search console** de google, celle des **keywords** n'est plus reconnue par Google suite aux abus de certains développeurs...

Extrait de mes balises <meta> incluse dans le <head> :

```
<meta name="viewport" content="width=device-width, initial-  
scale=1.0">  
<meta http-equiv="X-UA-Compatible" content="ie=edge">  
<title><%= title %></title>  
<meta name="description" content="Project Cars est un site  
d'annonces de véhicules d'occasions. Les voitures reviennent de fin  
de leasing crédit-bail elles disposent d'un faible kilométrage et  
d'un carnet d'entretien à jour" />  
<meta name="google-site-verification"  
content="GXs5KoUUkNCoaAZn7wPN-t01Pywp9M3sEjnt_3_ZWPc">
```

Attribut « alt » décrivant le contenu d'une image pour les malvoyants et les navigateurs non compatibles :

```

```

Sinon il y a le **référencement de payant SEA** avec l'**achat de mots clés** pour avoir un meilleur positionnement dans les résultats de recherche.

## RGPD

Le règlement n° 2016/679, dit règlement général sur la protection des données (**RGPD**, ou encore GDPR, de l'anglais *General Data Protection Regulation*), est un règlement de l'Union européenne qui constitue le texte de référence en matière de **protection des données** à caractère personnel. Il renforce et unifie la protection des données pour les individus au sein de l'Union européenne.

Dans ce projet j'ai appliqué certains principes de la RGPD ainsi lorsqu'un utilisateur souhaite se créer un compte, il doit préalablement cocher la case « **Accepter les CGU** », **cette case ne doit pas être cochée par défaut**.

J'ai donné à l'input de type checkbox l'attribut required pour qu'il soit obligatoirement coché par l'utilisateur, sans son **consentement** le formulaire ne peut être POSTER à la BDD, je n'ai pas défini l'attribut checked= « checked », car sinon la case aurait été automatiquement coché ce qui va à l'encontre de la RGPD :

```
<input type="checkbox" id="checkboxCGU" required>  
<label for="checkboxCGU">Veuillez accepter les CGU</label>
```

Pour assurer la confidentialité des données, j'ai attribué un rôle à chaque utilisateur, seul un administrateur ayant le statut de rôle 2 peut accéder à la base de données, j'ai appliqué un hachage sur les mots de passe, pour prendre un rendez-vous l'utilisateur aura un second formulaire pour communiquer son numéro de téléphone et une nouvelle checkbox à cocher.

user_id	user_email	user_password	user_firstname	user_lastname	user_phone	user_status
50	a@a.a	\$2b\$10\$0U.Svb GldNTxq/Npm VRMUeSUYoNt Mgv4xu4sloYae JP04BJyFWBSq	Wicem	SMONDEL	null	1

### 3. BACK-END

#### NODE.JS

**NODE.JS environnement** logiciel de bas niveau qui permet d'utiliser le langage JavaScript côté serveur et d'interroger de la base de données.

#### EXPRESS.JS

**EXPRESS.JS** est un **framework** **NODE.JS** qui permet de construire son architecture back-end plus aisément.

```
var express = require('express');
```

#### NODE MODULES

Les node modules sont des extensions que nous pouvons ajouter au noyau de Node.js, certains de ces modules sont déjà préinstallés, pour les autres nous utilisons **NPM** (Node Package Manager) comme outil pour télécharger ces paquets et les installer de façon locale ou globale. On retrouve la liste des dépendances utilisées au sein du fichier *package.json* :

```
"dependencies": {  
  "bcrypt": "^3.0.6",           //Password  
  "body-parser": "^1.19.0",     //Parse datas from body  
  "cookie-parser": "~1.4.4",  
  "debug": "~2.6.9",  
  "ejs": "~2.6.1",             //Views manager  
  "express": "~4.16.1",  
  "express-session": "^1.16.2", //Handle sessions  
  "http-errors": "~1.6.3",     //Display http errors  
  "morgan": "~1.9.1",  
  "mysql": "^2.17.1",          //Driver for MySQL  
  "req-flash": "0.0.3"         //Render flash messages  
}
```





## ARCHITECTURE MVC

**Modèle / Vue / Contrôleur** ou **MVC** est un motif d'architecture logicielle destiné aux interfaces graphiques. Le motif est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs.

Un modèle (Model) contient les données à afficher.

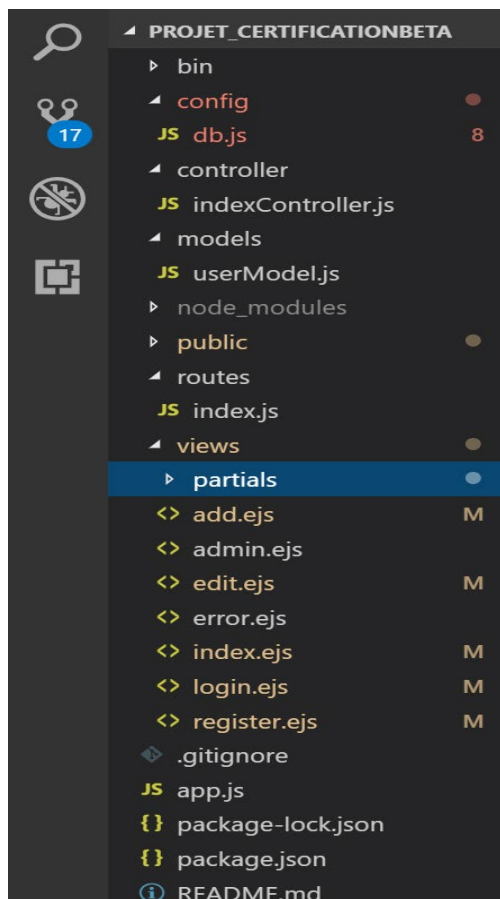
Une vue (View) contient la présentation de l'interface graphique.

Un contrôleur (Controller) contient la logique concernant les actions effectuées par l'utilisateur.

J'ai utilisé **express-generator** pour créer automatiquement mon architecture MVC avec EJS pour gérer les vues

```
wsmon@DESKTOP-UCAP5P8 MINGW64 ~/OneDrive/Bureau/ (master)
$ express projet_certificationBeta --view -ejs
```

Voici l'arborescence de mon projet :

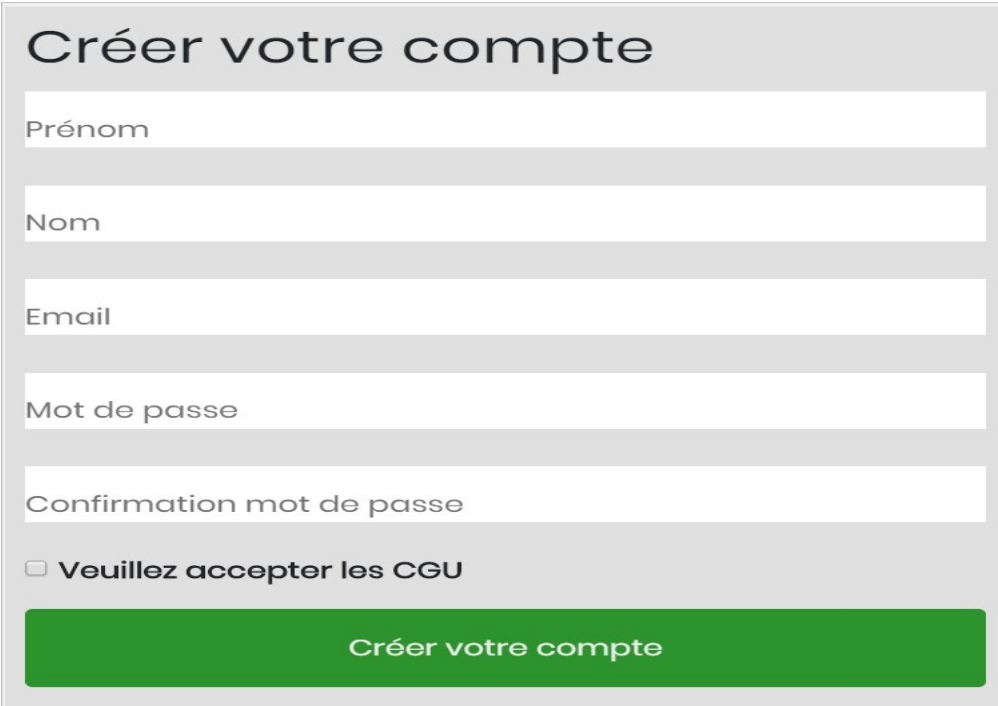


# FONCTIONNALITÉ REPRÉSENTATIVE

L'une des fonctionnalités les plus représentatives de mon projet « **Project Cars** » est celle de **l'authentification et de l'enregistrement des utilisateurs** pour laquelle j'ai effectué la recherche que je détaillerai dans le chapitre suivant.

Tout d'abord j'ai développé un **formulaire d'inscription** utilisant la méthode **POST** dans le fichier `/views/register.ejs`

```
<form class=" form-container" method="post" action="/register">
  <% if (typeof error !== 'undefined'){ %><h4 style="color:red">
    <%= error%></h4><% } %>
  <h2>Créer votre compte</h2>
  <input type="text" placeholder="Prénom" name="firstname" required>
  <input type="text" placeholder="Nom" name="lastname" required>
  <input type="email" placeholder="Email" name="email" required>
  <input type="password" placeholder="Mot de passe" name="password"
required>
  <input type="password" placeholder="Confirmation mot de passe"
name="passwordConfirm" required>
  <input type="checkbox" id="checkboxCGU" required>
  <label for="checkboxCGU">Veuillez accepter les CGU</label>
  <button type="submit" value="Register" class="btn">Créer votre
compte</button>
</form>
```



The screenshot shows a web form titled "Créer votre compte" (Create your account). It contains five input fields: "Prénom" (First name), "Nom" (Last name), "Email", "Mot de passe" (Password), and "Confirmation mot de passe" (Confirm password). Below the password fields is a checkbox labeled "Veuillez accepter les CGU" (Please accept the terms and conditions). At the bottom is a green button with the text "Créer votre compte".

J'ai en même temps créé le **formulaire d'authentification** dans le fichier `/views/login.ejs` qui lui aussi utilise la méthode **POST** et qui comporte un lien renvoyant au formulaire d'inscription :

```
<form class=" form-container" method="POST">
  <% if (typeof error !== 'undefined'){ %><h4 style="color:red"><%=
error%></h4><% } %>
  <h2>Connectez-vous</h2>
  <input type="text" placeholder="Email" name="email" required>
  <input type="password" placeholder="Mot de passe" name="password"
required>
  <button type="submit" class="btn">Valider</button>
  <a href="/register" class="btn cancel" href>Pas encore de compte
?</a>
</form>
```



Connectez-vous

Email

Mot de passe

Valider

Pas encore de compte ?

La connexion du serveur à la base de données se fait en appelant le node module MySQL servant de driver au serveur MySQL et par ce script du fichier `/config/db.js` que j'exporte pour l'utiliser à la demande dans les autres fichiers de mon **MVC**. J'y définis les identifiants de connexion requis par le serveur MySQL de ma base de données :

```

const mysql = require('mysql');

const connection = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: '*****',
  database: 'project_cars',
});

console.log('db connected')
module.exports = connection;

```

Je conçois mes **modèles** comportant les **requêtes** accédant aux données voulues dans ma **base de données** en élaborant une relation entre les **colonnes** de ma BDD et aux **attributs** « **name** » des "**placeholders**" de mes formulaires.

Dans mes fonctions *adduser* & *getuser* j'insère des requêtes préparées (**prepared statement**) en leur déterminant pour valeur « ? » empêchant l'injection de scripts SQL malveillants :

```

var mysql = require('mysql');
var connection = require('../config/db');
//Add user to the database
exports.adduser = function(firstname, lastname, email, password, cb) {
  {
    connection.query('INSERT INTO project_cars.t_users SET ?', {
      user_firstname: firstname, user_lastname: lastname, user_email:
      email, user_password: password },
      function(err, result) {

        if (err) return cb(err, null);
        return cb(null, result)
      });
  }
}
//Check user from the database
exports.getuser = function(email, cb) {
  connection.query("SELECT user_email, user_password FROM
project_cars.t_users WHERE user_email = ?", [email], function(err,
result, fields) {
    if (err) throw err;
    return cb(null, result);
  });
}

```

Ensuite je fixe les **routes** qui utilisent les **méthodes** que j'ai définies dans mon **contrôleur** :

```
var express = require('express');
var router = express.Router();
var user = require('../models/userModel');
var indexController = require('../controller/indexController');
var session = require('express-session');
// GET pages
router.route('/login')
  .get(indexController.login)
  .post(indexController.login);

router.route('/register')
  .get(indexController.register)
  .post(indexController.register);

module.exports = router;
```

Dans le contrôleur je définis la logique concernant les actions effectuées par l'utilisateur, je commence par requérir le driver de **MySQL**, les modèles pour insérer des nouveaux utilisateurs et vérifier les existants, le node module **Express-session** pour créer des sessions utilisateur de connexion et le node module **Bcrypt** qui procédera au hachage du mot de passe d'un nouvel utilisateur dans la base de données et comparera dans la BDD le hash du mot de passe d'un utilisateur se connectant au site.

Dans la fonction « **login** » je définis qu'un utilisateur déjà authentifié sera **redirigé** vers la page d'accueil et que si en cas d'erreur de login des **messages** préviendront l'utilisateur que soit l'email d'identification n'existe pas ou soit que le mot de passe saisi est erroné.

Pour la fonction « **register** » si l'utilisateur est déjà connecté il sera redirigé vers la page d'accueil sinon **Bcrypt** compare le mot passe et la confirmation de mot de passe s'ils sont **identiques** le **mdp** sera **haché** et les champs du formulaire seront insérés dans la BDD et l'utilisateur sera redirigé vers l'accueil. Si les **mdp** sont **différents** un **message d'erreur** s'affichera.

```

var mysql = require('mysql');
var User = require('../models/userModel');
var session = require('express-session');
var bcrypt = require("bcrypt");

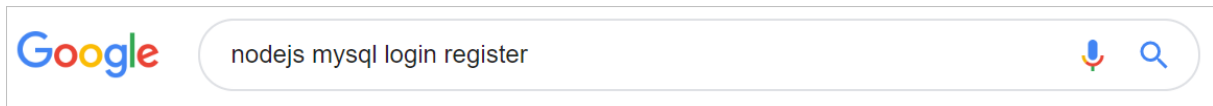
module.exports.login = function(req, res) {
  if (req.session.user)
    return res.redirect('/');
  if (req.method !== 'POST') res.render('login', { title: "Login" });
  else {
    if (req.body.email && req.body.password) {
      User.getUser(req.body.email, function(err, user) {
        if (err) res.render('login', { error: err, title: "Login" });
        else if (typeof user[0] == 'undefined') res.render('login', {
error: "Cet utilisateur n'existe pas !", title: "Login" });
        else {
          bcrypt.compare(req.body.password,
user[0].user_password).then(function(response) {
            if (response) {
              req.session.user = user;
              return res.redirect('/');
            } else {
              res.render('login', { error: 'Mot de passe
incorrect !', title: "Login" });
            }
          });
        }
      });
    }
  }
};

module.exports.register = function(req, res) {
  if (req.session.user)
    res.redirect('/');
  if (req.method !== 'POST') res.render('register', { title: "Register" });
  else {
    if (req.body.password === req.body.passwordConfirm) {
      bcrypt.hash(req.body.password, 10, function(err, hash) {
        User.adduser(req.body.firstname, req.body.lastname, req.body.email, hash,
          function(err, user) {
            if (err) res.render('register', { error: err, title: "Register" });
            else { return res.redirect('login'); }
          }
        );
      });
    } else res.render('register', { error: "Les mots de passe sont
différents !", title: "Register" });
  }
};

```

## SITUATION AYANT NÉCESSITÉ UNE RECHERCHE

Pour développer ma fonctionnalité d'authentification, j'ai fait la recherche suivante :



J'ai consulté les 4 premiers résultats :

### Basic Login System with Node.js, Express, and MySQL - CodeShack

<https://codeshack.io/basic-login-system-nodejs-express-mysql/> ▼ Traduire cette page

7 mai 2019 - Basic **Login** System with **Node.js**, Express, and **MySQL**. Updated on ... Establishing a **connection** to a **MySQL** database and selecting rows using **MySQL** queries. Creating GET and ..... **Login & Registration** system. Home page.

### Handling user login and registration using nodejs and mysql - Medium

<https://medium.com/.../handling-user-login-and-registration-using-...> ▼ Traduire cette page

29 janv. 2017 - In today's tutorial we are going to create basic email **login authentication** system using **nodejs** and **mysql**. If you need to set up **mysql** server on ...

### Creating Login and Registration Application Using NodeJS and MySQL!

[coding-karma.com > Javascript](https://coding-karma.com/.../javascript) ▼ Traduire cette page

12 août 2017 - Creating **Login** and **Registration** Application Using **NodeJS** and **MySQL**! Why choose Javascript Framework to interact with back-end server ...

### Node js User Authentication using MySQL and Express JS - JS-Tutorials

<https://www.js-tutorials.com/nodejs.../node-js-user-authentication-u...> ▼ Traduire cette page

Aller à **Login** and **Registration** page in **Nodejs** application - We will add **login** and **registration** page into this **nodejs** application, we will ...

J'ai décidé de suivre le tutoriel « Node js User Authentication using MySQL and Express JS » détaillé sur cette web :

<https://www.js-tutorials.com/nodejs-tutorial/node-js-user-authentication-using-mysql-express-js/>

« We will authenticate user using MySQL database. We will create GET and POST type HTTP request to show login and post login information to server. I am using Bootstrap CSS to create beautiful login and registration form.

*I am using following files and folder*

I am creating a folder '**nodejs\_auth\_example**'. This is our node js project name.

**views folder:** This folder will contain all ejs template files.

**public folder:** This folder will contain all external css and js files.

**routes folder:** This folder will contain all controller file.

**app.php:** This file will use to create nodejs application server and routes url.

**node\_modules folder:** This folder will contain all node.js packages.

### **Create Database and Table Structure for User Authentication**

I am using MySQL server for user authentication. You need to create '**test**' name database into mysql server. You can create database using GUI or from command line whatever you like, Lets create '**users**' table into '**test**' MySQL database using below script. »

## **TRADUCTION**

Nous authentifierons un utilisateur en utilisant une base de données MySQL. Nous créerons une requête http de type GET et POST pour montrer l'identifiant et poster les informations d'identification au serveur.

J'utilise les fichiers et dossiers suivants :

Je crée un dossier '**nojejs\_auth\_example**'. Ceci est le nom de notre projet node.js.

Dossier de vues : Ce dossier contiendra tous nos fichiers modèles EJS.

Dossier public : Ce dossier contiendra les fichiers CCS et JS externes.

Dossier routes : Ce dossier contiendra tous les fichiers du contrôleur.

App.php : Ce fichier servira à créer notre application serveur Node.js et les URL des routes.

Créer la structure d'une base de données et une table pour l'identification utilisateur.

J'utilise le serveur MySQL pour identifier un utilisateur. Vous avez besoin de créer une base de données nommée '**test**' au sein du serveur MySQL. Vous pouvez créer la base de données en utilisant une interface graphique ou un terminal de ligne de commandes selon votre préférence. Procédons à la création de la table '**users**' dans la base de données MySQL '**test**' en utilisant le script ci-dessous.



# VEILLE SUR LES VULNÉRABILITÉS DE SÉCURITÉ

Pour éviter l'**injection de requête SQL** malveillante j'ai fait une veille sur ce sujet en exécutant cette recherche sur Google :



J'ai consulté les 3 premiers résultats qui préconisaient tous les mêmes méthodes :

## How to prevent SQL Injection in Node.js - Hailstone.io

<https://blog.hailstone.io/how-to-prevent-sql-injection-nodejs/> ▼ Traduire cette page

9 juil. 2018 - While **Node.js** can help improve developer productivity, it's prone to **SQL injection** bugs as much as any other modern programming language ...

## How To Prevent SQL Injection In Node.js - Technicalkeeda.com

[www.technicalkeeda.com/nodejs.../how-to-prevent-sql-injection-in...](http://www.technicalkeeda.com/nodejs.../how-to-prevent-sql-injection-in-...) ▼ Traduire cette page

3 mars 2019 - How To **Prevent SQL Injection** In **Node.js**. require('mysql') - Load the mysql module to connect to database. To **avoid SQL Injection** attack, You need escape user input data before using it inside a **SQL** query. You can use mysql.escape() , connection.escape() or pool.escape() methods.

## Preventing SQL injection in Node.js - Stack Overflow

<https://stackoverflow.com/.../preventing-sql-injection-in-node-js> ▼ Traduire cette page

5 réponses

3 avr. 2013 - The **node-mysql** library automatically performs escaping when used as you are already doing.

J'ai suivi le tutoriel à l'adresse suivante :

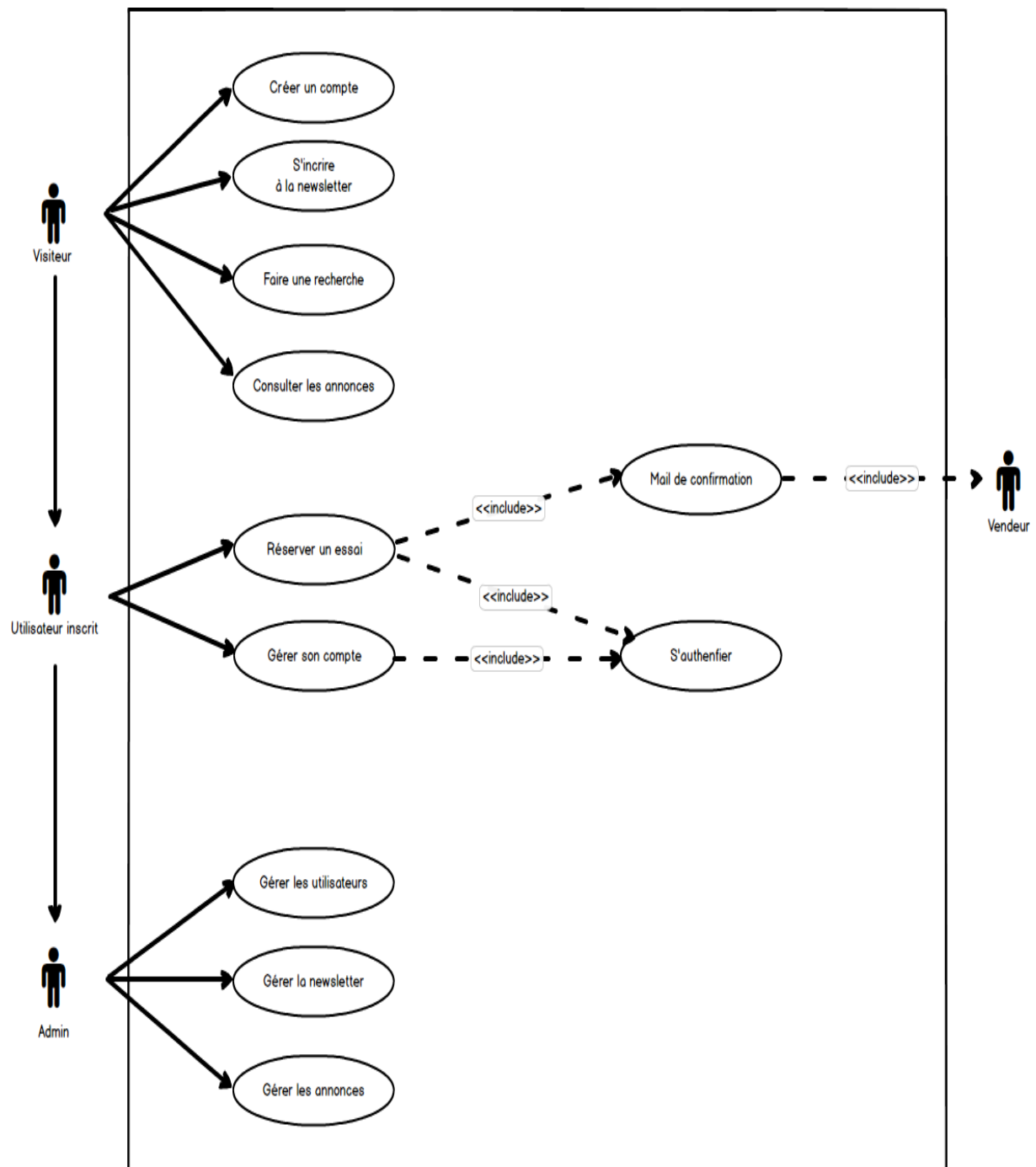
<https://blog.hailstone.io/how-to-prevent-sql-injection-nodejs/>

Et j'ai ainsi créé des **requêtes préparées** en combinant les 2 méthodes proposées en « *SET* » une valeur « *?* » et en créant un objet incluant les « **names** » des « **placeholders** » du formulaire.

```
connection.query('INSERT INTO project_cars.t_users SET ?', { user_firstname:
firstname, user_lastname: lastname, user_email: email, user_password: password
},
```

# ANNEXES N°1

## USE CASE



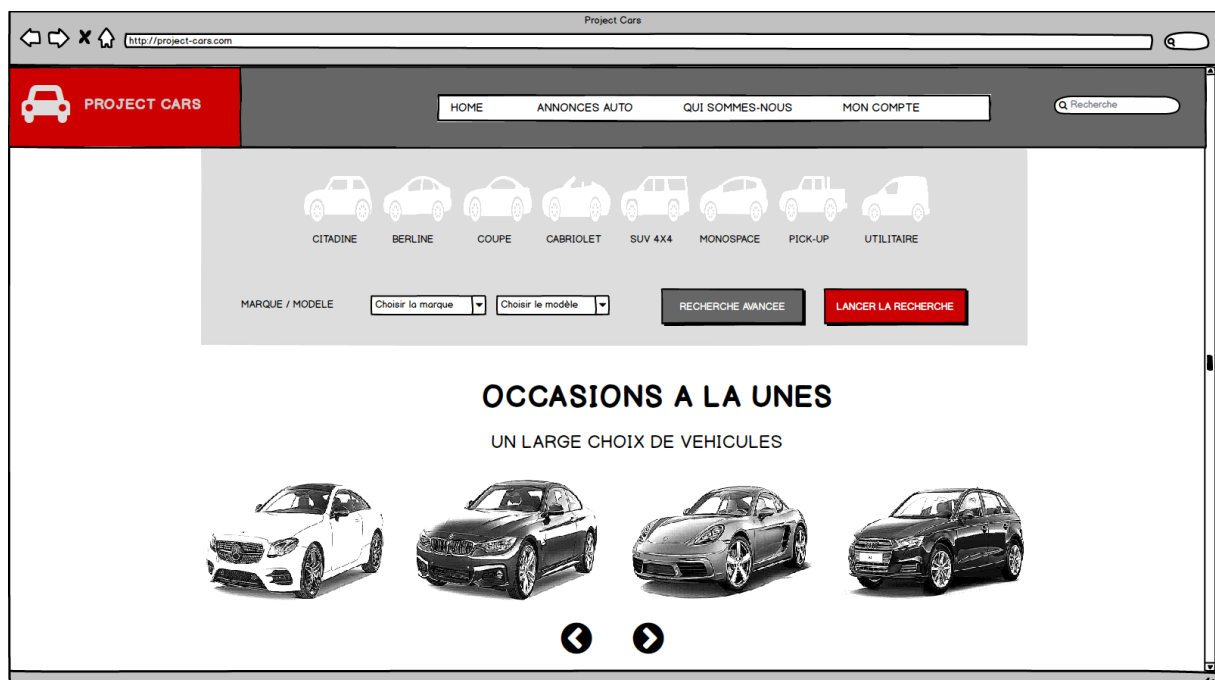
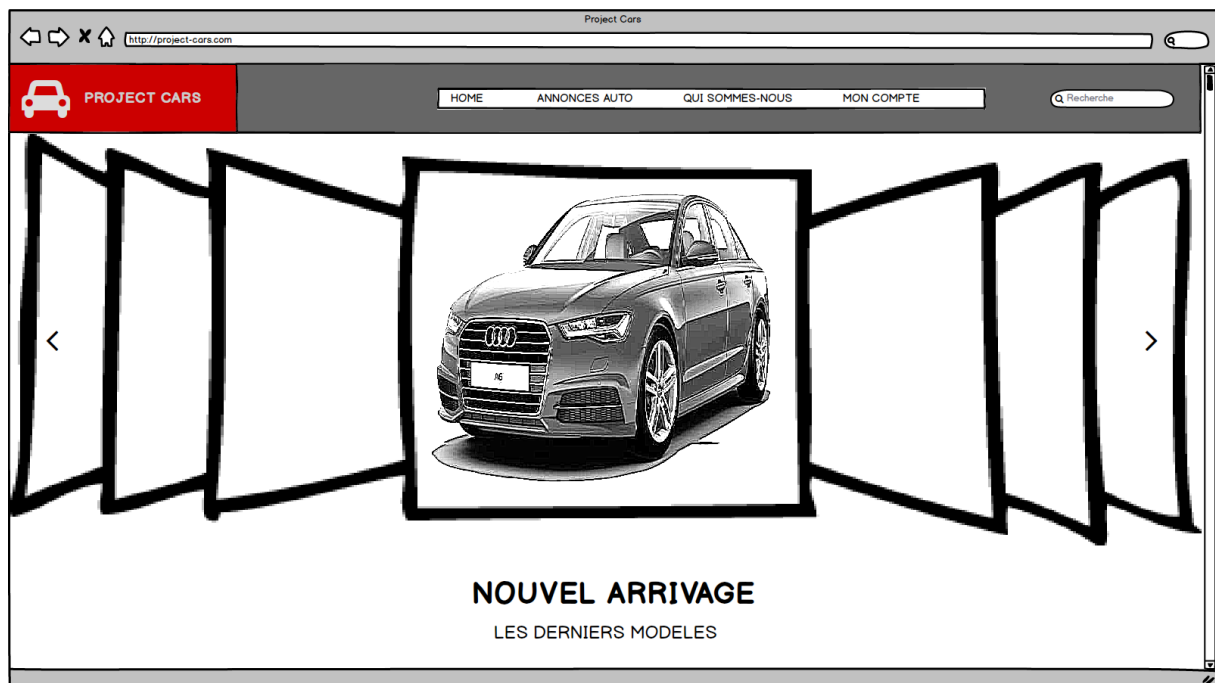
## ANNEXE N°2

### ARBORESCENCE DES PAGES WEB DU SITE PROJECT CARS

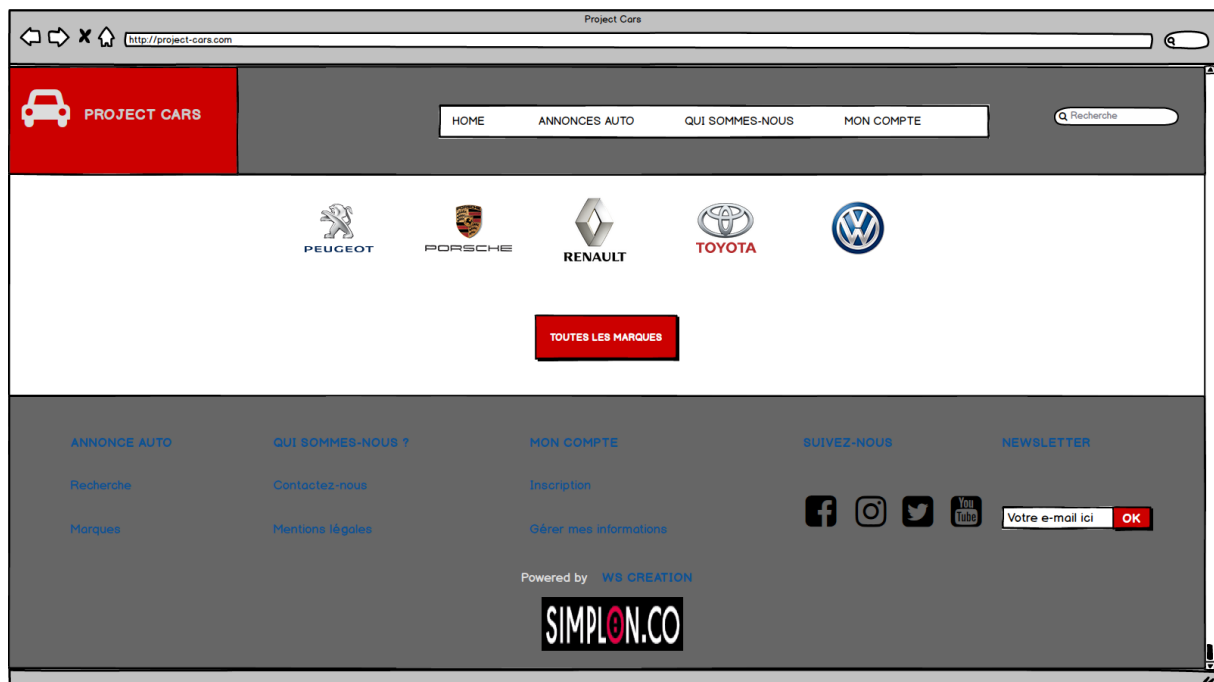
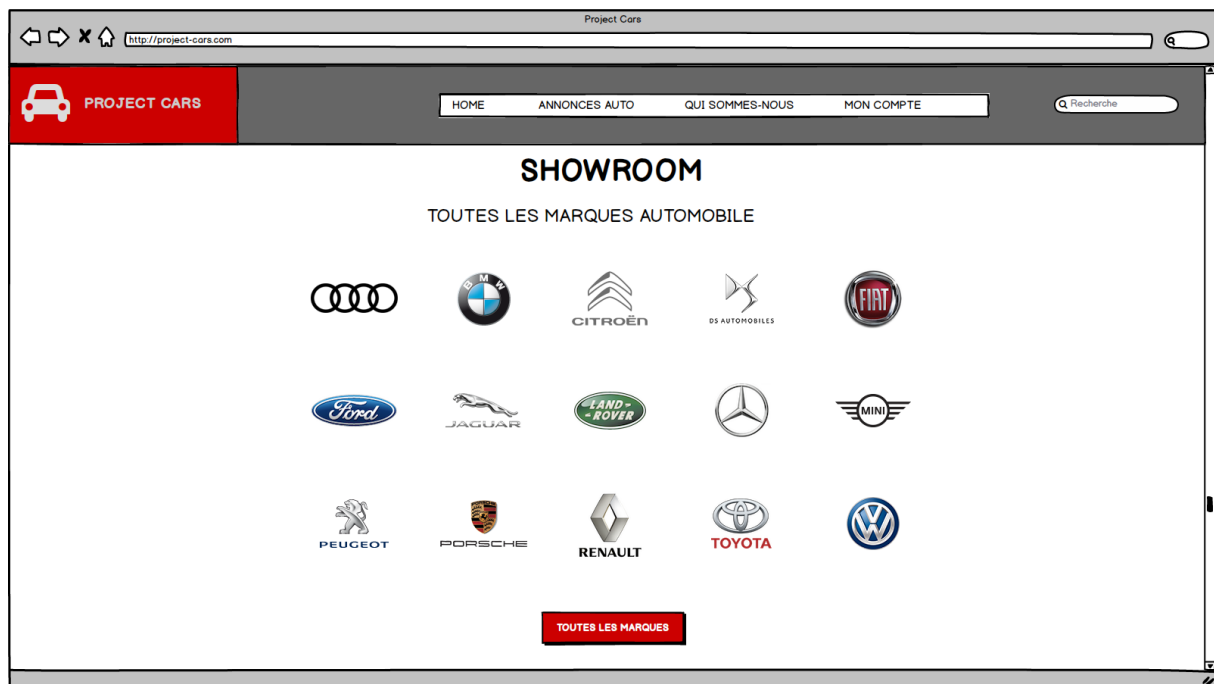


# ANNEXE N°3

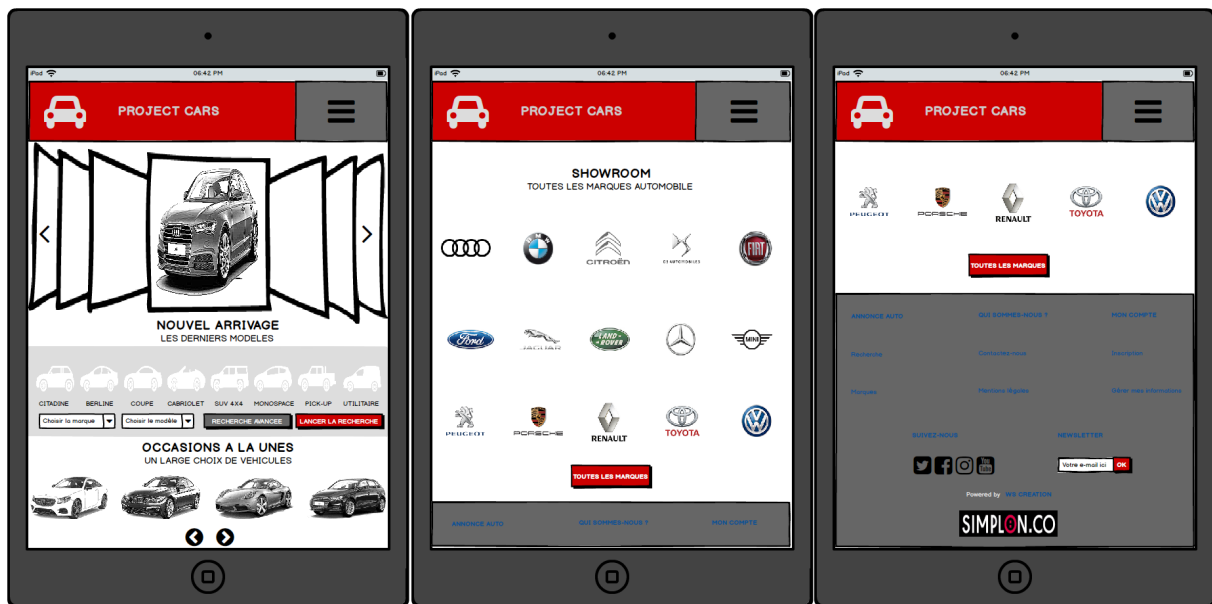
## 1. MAQUETTES DE LA LANDING PAGE VERSION DESKTOP



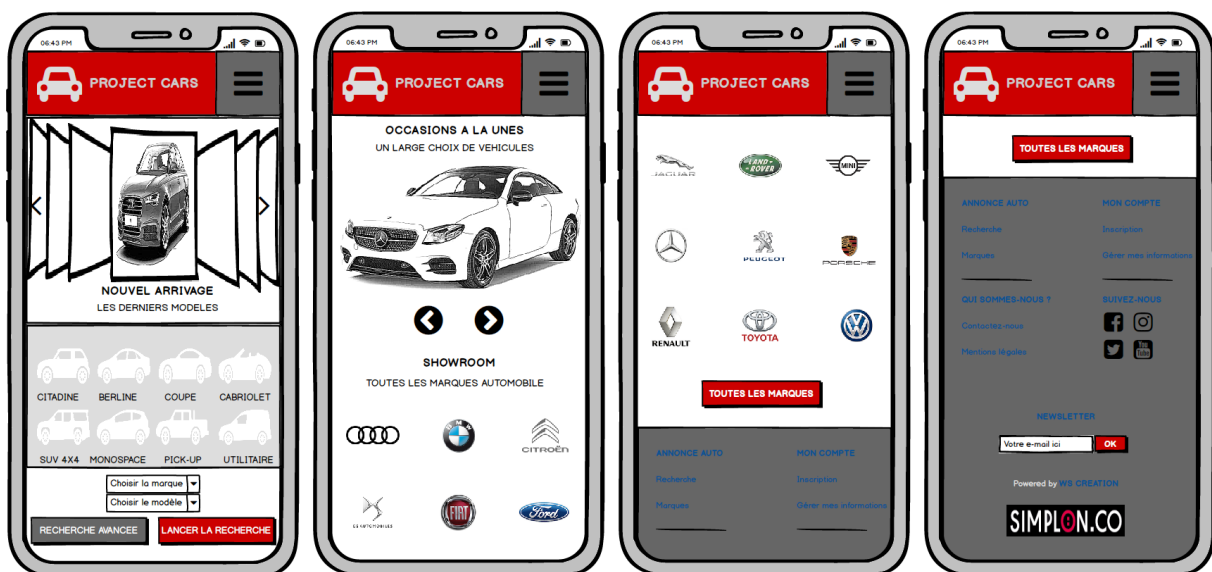
## 2. MAQUETTE DE LA LANDING PAGE VERSION PC (SUITE)



### 3. MAQUETTE DE LA LANDING PAGE VERSION TABLETTE



### 4. MAQUETTE DE LA LANDING PAGE VERSION MOBILE



# ANNEXE N°4

## MODELE CONCEPTUEL DE DONNEES

