# **RAPPORT DU PROJET**

# **CREATION D’UN SITE WEB DE VOITURE DE LUXE**

**Réalisé par :**

**Encadré par :**

* **M.EL ABID Noureddine**
* **Hajar KHOUZ**
* **Achraf MOUMEN**
* **Amine DEROUCIH**

# 

Année universitaire :

2023-2024

# REMERCIEMENTS

Tout d’abord, nous tenons à remercier tout particulièrement et à témoigner toute notre reconnaissance aux personnes suivantes, pour leur dévouement et leur soutien dans la concrétisation de ce projet :

* M. EL ABID AMRANI Noureddine, Professeur permanent à l’IGA, pour ses conseils éclairés, sa patience, sa disponibilité et pour la confiance qu’il nous a accordée dès l’ébauche du projet.
* L’IGA et l’ensemble des enseignants pour leur coopération professionnelle tout au long de cette expérience et pour avoir partagé avec nous, une partie de leurs savoir-faire et de leurs expériences professionnelles.

# Table des figures

[Figure 1 : Processus du planning du projet 8](#_Toc160018701)

[Figure 2 : Tableau de description des acteurs 10](#_Toc160018702)

[Figure 3 : Cas d’utilisation globale 14](#_Toc160018703)

[Figure 4 :Cas d’utilisation coté administrateur 15](#_Toc160018704)

[Figure 5 : Description du scénario nominal « ModifierPrixVoiture» 16](#_Toc160018705)

[Figure 6 :Diagramme de cas d’utilisation coté «client» 17](#_Toc160018706)

[Figure 7 : Description du scénario nominal « FaireReservation» 18](#_Toc160018707)

[Figure 8 :Diagramme de séquence d’authentification de l’admin du site 20](#_Toc160018708)

[Figure 9 :Diagramme de séquence d’ajout d’une voiture 21](#_Toc160018709)

[Figure 10 :Diagramme de séquence pour consulter la page d’acceuil 21](#_Toc160018710)

[Figure 11 :Diagramme de séquence de la réservation d’une voiture 22](#_Toc160018711)

[Figure 12 :Diagramme de classes du site 24](#_Toc160018712)

[Figure 13 : Architecture client-serveur 29](#_Toc160018713)

[Figure 14 :logo PHP 29](#_Toc160018714)

[Figure 15 :logo MySQL 30](#_Toc160018715)

[Figure 16 :logo html 31](#_Toc160018716)

[Figure 17 : logo CSS 31](#_Toc160018717)

[Figure 18 :logo JavaScript 32](#_Toc160018718)

[Figure 19 : logo Bootstrap 32](#_Toc160018719)

[Figure 20 : logo XAMPP 33](#_Toc160018720)

[Figure 21 : logo Firefox 33](#_Toc160018721)

[Figure 22 : logo PowerAMC 34](#_Toc160018722)

[Figure 23 : logo Visual Studio Code 34](#_Toc160018723)

[Figure 24 : Page d’accueil 37](#_Toc160018724)

[Figure 25 : Formulaire de réservation 38](#_Toc160018725)

[Figure 26 : Les voitures disponibles pour la réservation 38](#_Toc160018726)

# Table des matières

[REMERCIEMENTS 1](#_Toc160019431)

[Table des figures 2](#_Toc160019432)

[Table des matières 3](#_Toc160019433)

[INTRODUCTION 5](#_Toc160019434)

[CHAPITRE 1 : Contexte général du projet 8](#_Toc160019435)

[CHAPITRE 2 : ANALYSE ET CONCEPTION 11](#_Toc160019436)

[CHAPITRE 3 : Langages et Outils de Développement 28](#_Toc160019437)

[CHAPITRE 4 : Réalisation du projet 37](#_Toc160019438)

[CONCLUSION GENERALE 75](#_Toc160019447)

[Bibliographie 77](#_Toc160019448)

# INTRODUCTION

Dans un monde où le luxe rencontre la technologie, le marché des voitures de prestige a atteint de nouveaux sommets d’innovation et d’exclusivité.

Dans ce paysage dynamique et en constante évolution, développer un site Web dédié aux voitures de luxe constitue une opportunité inestimable de combiner l’art du design automobile avec la puissance de la connectivité en ligne.

Ce rapport détaille le projet de création d'un site Web sur les voitures de luxe, conçu pour capturer l'essence de l'exclusivité, de la performance et de la sophistication.

Notre objectif est de créer une plateforme en ligne riche et premium qui offre aux passionnés d'automobile une expérience sans précédent dans leur quête de perfection.

Ce projet ambitieux repose sur plusieurs piliers fondamentaux : une sélection minutieuse des marques et des modèles présentés, une conception d'interface utilisateur élégante et intuitive et l'intégration de fonctionnalités innovantes pour offrir une expérience utilisateur incomparable.

Ce présent rapport résume l’ensemble des étapes que j’ai pu effectuer pour la mise en place de l’application intitulée : réalisation d’un site web pour la réservation de voitures de luxe. Il est organisé, en plus de cette introduction, en quatre chapitres et une conclusion.

Le premier chapitre est réservé pour la présentation du contexte général du projet.

Le deuxième chapitre sera consacré à la partie analyse et conception de l’application.

Dans le troisième chapitre nous présentons l’ensemble des interfaces du site web réalisé.

Dans le quatrième chapitre nous présentons le projet réalisé

Une conclusion sera donnée en fin de ce rapport.

Ce projet vise à établir une norme pour la présentation en ligne de voitures de prestige, tout en offrant aux passionnés une plateforme unique pour apprendre et explorer.

# CHAPITRE 1 : Contexte général du projet

**I. Présentation du Projet**

Le projet de création d'un site web dédié aux voitures de luxe vise à établir une plateforme en ligne haut de gamme qui offre une expérience immersive aux passionnés et acheteurs potentiels de voitures de luxe. Notre objectif principal est de créer un environnement digital sophistiqué qui reflète le prestige et l'exclusivité associés aux marques de voitures de luxe.

**I.1 Étude de l'Existant**

Avant de débuter la conception et le développement du site web, une analyse approfondie de l'existant dans le domaine des sites web automobiles et des plateformes de vente en ligne de voitures de luxe a été réalisée. Cette étude a permis de comprendre les tendances du marché, les pratiques des concurrents, ainsi que les attentes et préférences des clients potentiels.

**I.2 Étude Générale du Projet**

**I.2.1 Cahier des Charges**

Le cahier des charges du projet détaille les fonctionnalités, les caractéristiques et les exigences spécifiques du site web de voitures de luxe. Il inclut notamment la conception de l'interface utilisateur, les fonctionnalités de recherche avancée, le processus de commande et de paiement sécurisé, la gestion des comptes utilisateurs, ainsi que les aspects liés au référencement et à la visibilité sur les moteurs de recherche.

**I.2.2 Problématique et Solutions**

La problématique centrale du projet réside dans la création d'un site web qui parvient à captiver et à engager les utilisateurs tout en reflétant l'essence même du luxe et de l'exclusivité associés aux voitures de prestige. Pour résoudre cette problématique, nous avons opté pour une approche centrée sur l'esthétique, la performance et l'expérience utilisateur. Nous mettons également l'accent sur la personnalisation et la segmentation du contenu pour offrir une expérience sur mesure à chaque visiteur.

**I.2.3 Planning du Projet**

Le planning du projet est établi en plusieurs étapes clés, incluant la conception initiale, le développement du site web, les phases de test et de validation, ainsi que le déploiement et la maintenance. Chaque étape est minutieusement planifiée afin de garantir la cohérence et la qualité du résultat final, tout en respectant les délais et les contraintes budgétaires définis.

Figure 1 : Processus du planning du projet

En somme, le projet de création d'un site web de voitures de luxe repose sur une analyse approfondie du marché et des besoins des utilisateurs, ainsi qu'une approche méthodique et structurée pour concevoir et développer une plateforme en ligne qui incarne le luxe, l'exclusivité et l'excellence.

# CHAPITRE 2 : ANALYSE ET CONCEPTION

**II. 1 Définition du Projet**

**II.1.1Objectifs du projet**

Ce site web a plusieurs objectifs parmi lesquels on peut citer :

* Permettre à un visiteur de passer de l’état utilisateur à l’état client.
* Permettre d’effectuer une réservation en ligne.

**II.1.2 Identification des acteurs**

Le tableau ci-dessous représente la liste des acteurs que contient le projet :

|  |  |
| --- | --- |
| **ACTEUR** | **DESCRIPTION** |
| Administrateur | Gère les réservations, les comptes et les prix de location. |
| Utilisateur | Représente le visiteur du site web. |
| Client | Le visiteur possédant un compte  Client. |

Figure 2 : Tableau de description des acteurs

**II.1.3 Contraintes à respecter**

* Logo : proposer un logo qui est basé sur le nom de l’agence et son secteur d’activité, et qui sera en même temps simple et percutant
* L’ergonomie : avec l’augmentation exponentielle du nombre de tablettes et de smartphones surfant sur internet, il est indispensable de prévoir l’affichage du site sur ces supports. L’affichage peut être différent selon le support pour s’adapter à la taille de l’écran et au mode d’utilisation des supports : souris, stylet, doigts.
* Un contenu cohérent et de qualité : cela passe par la qualité de l’écriture (le style), le respect de l’orthographe et de la grammaire, la qualité des images et leur force illustrative, l’originalité et la simplicité
* La sécurité : l’accès aux informations de la base de données doit être sécurisé, seuls l’admin aura la possibilité d’y accéder.

**II. 2 Présentation d’UML :**

**II.2.1 Définition d’UML :**

Le langage UML (Unified Modeling Language, ou langage de modélisation unifié) a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Il est destiné à l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement. L'UML a des applications qui vont au-delà du développement logiciel, notamment pour les flux de processus dans l'industrie.

Il ressemble aux plans utilisés dans d'autres domaines et se compose de différents types de diagrammes. Dans l'ensemble, les diagrammes UML décrivent la limite, la structure et le comportement du système et des objets qui s'y trouvent.

L'UML n'est pas un langage de programmation, mais il existe des outils qui peuvent être utilisés pour générer du code en plusieurs langages à partir de diagrammes UML. L'UML a une relation directe avec l'analyse et la conception orientées objet.

**II.2.2 Choix de la méthodologie :**

Nous avons choisi UML pour ces raisons:

✔ Les digrammes UML supportent l'abstraction. Leur niveau de détail caractérise le niveau d'abstraction du modèle.

✔ La structure des diagrammes UML et la notation graphique des éléments de modélisation sont normalisées.

Le recourt à des outils appropriés est un gage de productivité pour la rédaction des diagrammes UML, car:

* Ils facilitent la navigation entre les différentes vues.
* Ils permettent de centraliser, organiser, partager et synchroniser les diagrammes (indispensables avec un processus itératif),

■ Facilitent l'abstraction, par des filtres visuels,

■ Simplifient la production de documents et autorisent (dans certaines limites) la génération de code.

**II.2.3 Les diagrammes d'UML :**

UML 2.0 comporte ainsi treize types de diagrammes représentant autant de vues

Distinctes pour représenter des concepts particuliers du système d'information. Ils se répartissent

en deux grands groupes:

-Diagrammes structurels ou diagrammes statiques (UML Structure) :

✓ diagramme de classes (Class diagram)

✓ diagramme d'objets (Object diagram)

✓ diagramme de composants (Component diagram)

✓ diagramme de paquetages (Package diagram)

✓ diagramme de structures composites (Composite structure diagram)

Diagrammes comportementaux ou diagrammes dynamiques (UML Behavior) :

✔diagramme de cas d'utilisation (Use case diagram)

✓ diagramme d'activités (Activity diagram)

✓ diagramme d'états-transitions (State machine diagram)

✓ Diagrammes d'interaction (Interaction diagram)

✓ diagramme de séquence (Sequence diagram)

Ces diagrammes, d'une utilité variable selon les cas, ne sont pas nécessairement tous produits à l'occasion d'une modélisation. Les plus utiles pour la maîtrise d'ouvrage sont les diagrammes de cas d'utilisation, de classes et de séquence.

**II.2.4 Phase de Conception :**

**II.2.4 .1 Diagramme de cas d’utilisation :**

Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. C'est le premier diagramme du modèle UML, celui où s'assure la relation entre l'utilisateur et les objets que le système met en œuvre.

Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous- système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit. Il scinde la fonctionnalité du système en unités cohérentes, les cas d'utilisation, ayant un sens pour les acteurs. Les cas d'utilisation permettent d'exprimer le besoin des utilisateurs d'un système, ils sont donc une vision orientée utilisateur de ce besoin au contraire d'une vision informatique.

Il ne faut pas négliger cette première étape pour produire un logiciel conforme aux attentes des utilisateurs. Pour élaborer les cas d'utilisation, il faut se fonder sur des entretiens avec les utilisateurs. Ce diagramme se compose de deux éléments :

✓ Acteur : Un acteur est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système. Il se représente par un petit bonhomme.

✓ Cas d'utilisation : Un cas d'utilisation est une unité cohérente représentant une fonctionnalité visible de l'extérieur. Un cas d'utilisation modélise donc un service rendu par le système, sans imposer le mode de réalisation de ce service.

Dans notre cas le système de réservation de voitures de luxe en ligne possède deux acteurs : le client (internaute) et le webmaster (administrateur du site) et un ensemble des cas d'utilisations qui représente les différentes fonctionnalités du système.

**Ce paragraphe représente une description détaillée des besoins fonctionnels mentionnés en bref dans le chapitre I partie des objectifs du projet :**

* **Permettre au client de s’identifier afin de pouvoir effectuer une réservation.**
* **Permettre à l’administrateur de modifier les prix des voitures.**
* **Permettre à l’administrateur de s’authentifier et de consulter les réservations effectuées par les clients.**
* **Permettre à l’administrateur de vérifier chaque réservation afin de l’accepter ou la rejeter dans le cas où la voiture souhaitée ne sera pas disponible pour les dates précisées dans la réservation.**
* **Permettre d’enregistrer les messages et les réservations dans la base de données.**

Après avoir déterminé les différents acteurs, nous avons procédé à la modélisation du comportement du site vis-à-vis des différents acteurs :

Une image contenant diagramme, croquis, ligne, cercle

Description générée automatiquement

Figure 3 : Cas d’utilisation globale

Nous allons par la suite détailler les deux diagrammes de cas d’utilisation relatifs au client et à l’administrateur.

**II.3 Description des packages du système**

**II.3.1** **Package administrateur**

Pour ce package le seul acteur c’est l’administrateur à qui on va attribuer un login et un mot de passe.

Le package administrateur a toutes les fonctionnalités, il a le droit de gérer les voitures (modifier les prix de locations des voitures disponibles, ajouter une nouvelle voiture, supprimer une voiture).

**II.3.1.1** **Diagramme des cas d’utilisation**

Une image contenant dessin, croquis, texte, diagramme

Description générée automatiquement

Figure 4 :Cas d’utilisation coté administrateur

**II.3.1.2** **Description des cas d’utilisation du package « admin»**

**II.3.1.2.1 Cas d’utilisation «ModifierPrixVoiture»**

* Titre : ModifierPrixVoiture.
* Résumé : cette étape permet à l’administrateur de modifier le prix de location des voitures.
* Acteurs : l’administrateur.
* Pré-conditions : avoir un login et un mot-de-passe correctes.
* Action déclencheur : l’administrateur souhaite modifier le prix d’une voiture.
* Scénario nominal :

|  |  |
| --- | --- |
| Action du(des)acteur(s) | Acteur du système |
| 1.L’acteur demande d’afficher les voitures et leurs prix. |  |
|  | 2.Le système affiche les prix de toutes les voitures enregistrées dans la base de données. |
| 3. L’acteur modifie le prix d’une  voiture. |  |
|  | 4. Le système enregistre les  modifications. |
|  | 5. Le système affiche un message de  réussite de la modification. |

Figure 5 : Description du scénario nominal « ModifierPrixVoiture»

**II.3.2**  **Package client**

Ce package consiste effectuer une réservation, les acteurs sont les clients qui possèdent un compte.

L’utilisateur qui est inscrit dans la base de données à la possibilité d’effectuer une réservation, ainsi d’envoyer des réclamations, c’est ce qu’on va représenter dans le diagramme des cas d’utilisation ci-dessous :

**II.3.2.1** **Diagramme des cas d’utilisation**

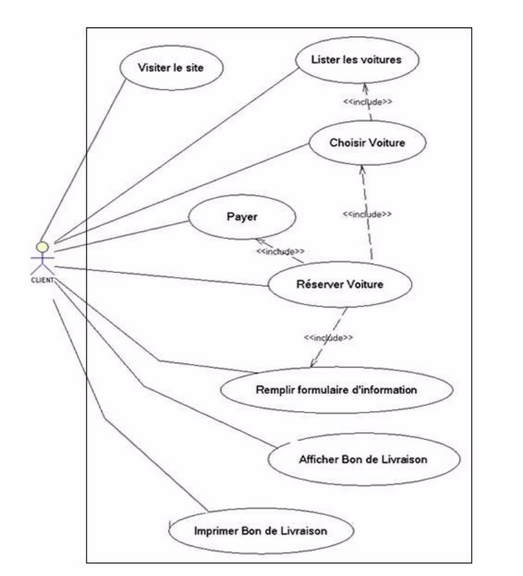


Figure 6 : Diagramme de cas d’utilisation coté «client»

**II.3.2.2** **Description des cas d’utilisation du package « client»**

**II.3.2.2.1 Cas d’utilisation «FaireReservation»**

* Titre : FaireReservation.
* Résumé : cette étape permet à l’utilisateur, qui est déjà inscrit dans la base de données, d’effectuer une réservation.
* Acteurs : les clients.
* Pré-conditions : avoir un compte client.
* Action déclencheur : le client souhaite effectuer une réservation en ligne.
* Scénario nominal :

|  |  |
| --- | --- |
| **Action du (des) acteur(s)** | **Acteur du système** |
| 1. L’acteur choisis la voiture à  réserver. |  |
|  | 2. Le système affiche les informations  de cette voiture. |
| 3. L’acteur complète les informations  de la réservation. |  |
| 4. L’acteur confirme sa réservation. |  |
|  | 5. Le système affiche un message de  réussite d’envoi au client. |

Figure 7 : Description du scénario nominal « FaireReservation»

**II.2.4 .2 Diagramme de séquence:**

Le diagramme de séquence montre des interactions entre objets selon un point de vue temporel. Ce type de diagramme sert à modéliser les aspects dynamiques des systèmes temps réels et des scénarios complexes mettant en œuvre peu d'objets.

Dans ce type de diagrammes, l'accent est mis sur la chronologie des envois de messages. La représentation se concentre sur l'expression des interactions et non pas sur l'état ou le contexte des objets.

Ce type de diagramme est usuellement utilisé pour illustrer les diagrammes de cas d'utilisation.

Dans ce qui suit, nous présenterons les scénarios des principaux cas d'utilisations des deux côtés : côté client et côté Webmaster.

**II.2.4.2.1** **Package administrateur :**

**II.2.4.2.1.1 Diagramme de séquence:**

Pour que l’admin gère les différents stockages de site, il faut s’authentifier :

Une image contenant texte, diagramme, ligne, Parallèle

Description générée automatiquement

Figure 8 : Diagramme de séquence d’authentification de l’admin du site

Ce diagramme décrit les différentes tâches que l’admin va suivre pour ajouter une voiture à leur BDD :

Une image contenant texte, diagramme, ligne, Parallèle

Description générée automatiquement

Figure 9 : Diagramme de séquence d’ajout d’une voiture

**II.2.4.2.2** **Package client :**

**II.2.4.2.2.1 Diagramme de séquence:**

Une image contenant texte, capture d’écran, ligne, diagramme

Description générée automatiquement

Figure 10 : Diagramme de séquence pour consulter la page d’accueil

Une image contenant texte, diagramme, Parallèle, ligne

Description générée automatiquement

Figure 11 : Diagramme de séquence de la réservation d’une voiture

**II.2.4 .3 Diagramme de classes :**

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation.

Alors que le diagramme de cas d'utilisation montre un système du point de vue des acteurs, le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d'utilisation.

Il s'agit d'une vue statique car on ne tient pas compte du facteur temporel dans le comportement du système.

Les principaux éléments de cette vue statique sont les classes et leurs relations: association, généralisation et plusieurs types de dépendances, telles que la réalisation et l'utilisation.

✓ Les classes: Une classe est la description formelle d'un ensemble d'objets ayant une sémantique et des caractéristiques communes.

✓ Relations entre classes: Une association est une relation entre deux classes (association binaire) ou plus (association n-aire), qui décrit les connexions structurelles entre leurs instances. Une association indique donc qu'il peut y avoir des liens entre des instances des classes associées.

Après l’étude de notre système un ensemble d’objets qui ont été recensés permettant en effet de dégager le diagramme de classes. Dans ce qui suit **la figure 12** présentera le diagramme de classes de notre site.

Une image contenant texte, diagramme, ligne, Parallèle

Description générée automatiquement

Figure 12 : Diagramme de classes du site

**III. Conclusion :**

Dans ce chapitre nous avons présenté les exigences et les besoins des différentes protagonistes ainsi que leur interactions avec le système et avec les utilisateurs à travers les cas d’utilisation et des scénarios pour mieux comprendre le rôle, les actions et les droits de chaque acteur dans notre site Web.

# CHAPITRE 3 : Langages et Outils de Développement

﻿

**1. Présentation du Web :**

**1.1. Site web :**

Ensemble de pages web hyperliées entre elles appartenant à un même serveur web et mises en ligne à une adresse Web.

**1.2. Qu'est-ce qu'un site statique :**

On entend par page statique, non pas une page sans mouvements ou sans animations, mais Page visible telle qu'elle a été conçue.

Ces pages peuvent présenter toute forme de contenu, animations flash, images, musique, vidéo etc... Mais elles sont toujours présentées de la même façon. Elles ne changent pas et c'est en ce sens qu'elles sont statiques.

**1.3. Qu’est-ce qu'un site dynamique :**

En opposition aux pages statiques, les pages dynamiques permettent de présenter les informations de différentes manières selon l'interaction avec le visiteur. Les pages sont alors construites "à la volée" grâce à une programmation conçue par le webmaster le contenu est issu d’une base de données en fonction de critères établis par l’internaute.

**1.3. Comment choisir si une page est statique ou dynamique ?**

Tout d’abord il convient de déterminer le rôle de la page. Tout système d’interaction avec le visiteur nécessitera une partie dynamique. Pour un système de mise à jour, il conviendra de réfléchir aux fréquences des mises à jour. Si le contenu ne change peu ou qu'une fois à l'année, il n'est pas nécessaire de passer par des pages dynamiques. Une intervention ponctuelle du webmaster peut très bien convenir.

**1.5. Quels sont les avantages et les inconvénients des deux méthodes?**

L'utilisation des pages dynamiques est plus souple pour l'évolution du site mais nécessite

Plus de temps de travail pour la programmation.

En opposition, les pages statiques sont plus "figées" mais plus simples à réaliser.

**1.6. Page web :**

Ressource du World Wide Web conçue pour être consultée par des visiteurs à l'aide d'un navigateur Web.

**1.7. Hyperlien ou lien hypertexte:**

Référence dans un système hypertexte permettant de passer automatiquement d'un document consulté à un document lié.

﻿

**1.8. Adresse Web ou URI (URL) :**

Référence universelle d'une ressource sur le web (et plus encore), de la forme :

**protocole://serveur:port/chemin#ancre?params**

**1.9. Serveur Web:**

Ensemble machine et logiciel hébergeant des pages web

**2. Comment créer une application Web avec PHP/MySQL...?**

De nombreuses applications Web reposent sur l'utilisation de bases de données (paniers de commandes, forums, ...).

Les bases de données sont utilisées dans les applications WEB pour permettre de stocker des données nécessaires à l'application. Si par exemple un site vitrine de type site d'entreprise n'aura nul besoin de base de données, car il ne sert qu'à présenter des informations, un site marchand devra stocker ses produits dans une base de données et mettre à jour les quantités pour un produit, en dynamique, après l'achat effectué par un de ses clients.

Une application Web en PHP/MySQL repose sur une architecture client-serveur mettant en jeu trois acteurs (Figure 13):

✔un serveur HTTP (ou serveur Web) permettant l'exécution de scripts PHP,

✓ Un serveur de bases de données MySQL,

✓ Un client http (navigateur),

Le client émet une requête HTTP. Le serveur Web la reçoit et transmet une réponse au client. Lorsque la réponse nécessite l'interrogation d'une base de données, une interface de programmation est utilisée pour communiquer avec la base.

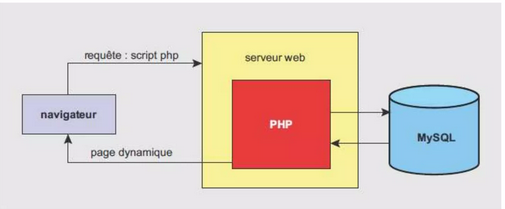


Figure 13 : Architecture client-serveur

﻿

**2.1. Qu’est-ce que PHP?**

PHP est un l'un des langages de script les plus actifs sur le Web. Il permet de créer principalement des pages Web HTML mais aussi d'autres types de contenu comme des images, des animations Flash, des documents PDF, etc.

PHP peut être utilisé soit comme un langage de script répondant à des besoins simples et à court terme (c'est ce qui l’a fait connaître), soit comme un langage de programmation complexe permettant de mettre en place des applications d’entreprise (programmation orientée objet, design patterns etc.).



Figure 14 :logo PHP

**2.2. Qu’est-ce que MySQL?**

SQL signifie«< Structured Query Language >> c'est-à-dire Langage d'interrogation structuré. En fait SQL est un langage complet de gestion de bases de données relationnelles. Il a été conçu par IBM dans les années 70. Il est devenu le langage standard des systèmes de gestion de bases de données (SGBD) relationnelles(SGBDR).

C'est à la fois :

✔un langage d'interrogation de la base (ordre SELECT)

✓ Un langage de manipulation des données (LMD; ordres UPDATE, INSERT, DELETE)

✓un langage de définition des données (LDD; ordres CREATE, ALTER, DROP),

✓un langage de contrôle de l'accès aux données (LCD; ordres GRANT, REVOKE). Le langage SQL est utilisé par les principaux SGBDR : DB2, Oracle, Informix, Ingres, RDB,... Chacun de ces SGBDR a cependant sa propre variante du langage. Ce support de cours présente un noyau de commandes disponibles sur l'ensemble de ces SGBDR, et leur implantation dans Oracle Version 7.

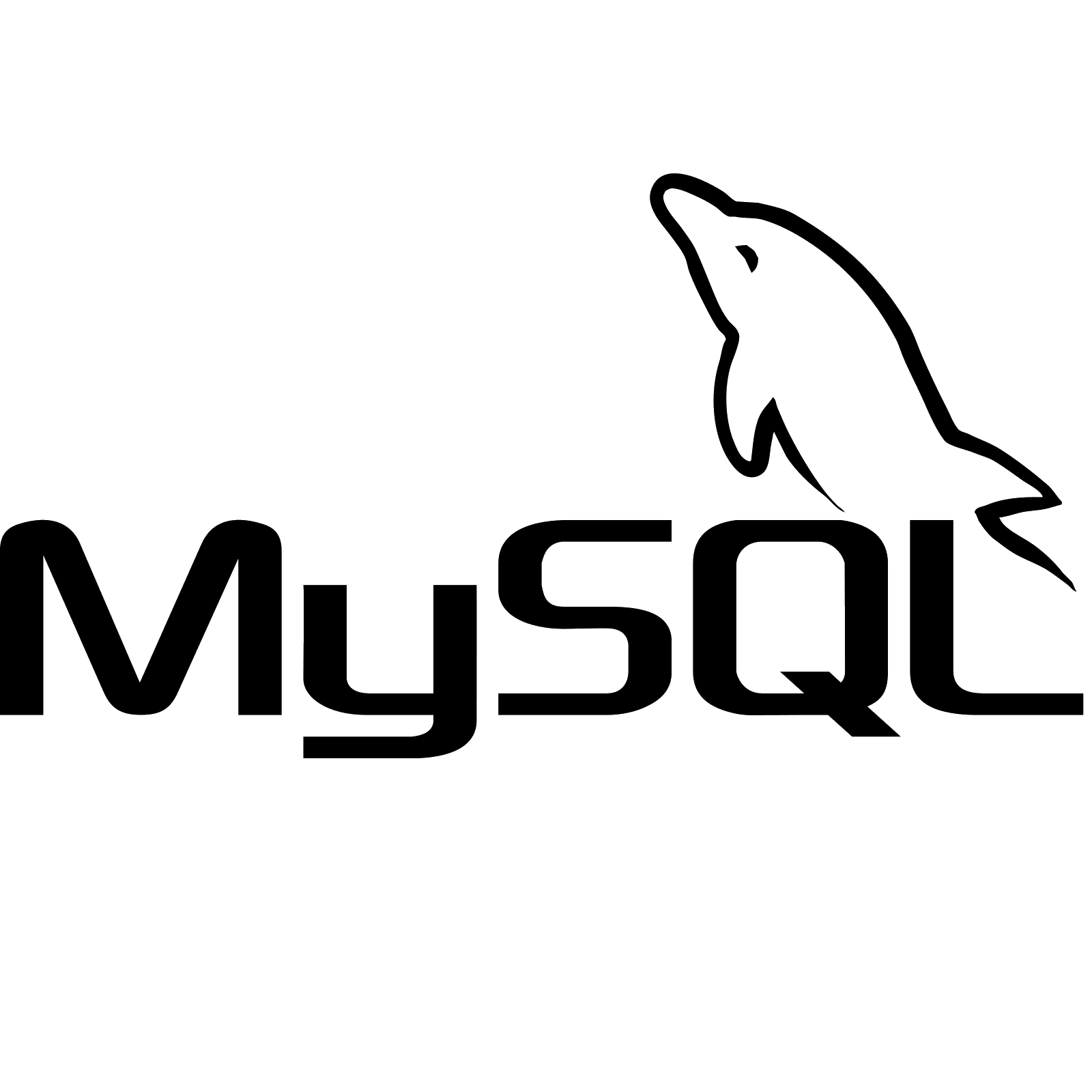


Figure 15 :logo MySQL

**2.3. Le langage HTML :**

HTML est le langage universel utilisé pour communiquer sur le Web. Votre information sera ainsi transportée sur cette gigantesque toile de réseaux interconnectés qu'est Internet, pour ﻿aboutir sur l'ordinateur de votre lecteur grâce à un programme appelé navigateur ou browser. Le logiciel que l'on appelle un browser vous permet de surfer sur le Net et d'afficher sur votre écran les "pages" qu'il a interceptées. Il y a, hélas, beaucoup de marques et de types de browsers différents. Des simples, des archaïques ou des sophistiqués... Les plus connus sont Netscape Navigator et Internet Explorer de Microsoft mais il en existe beaucoup d'autres. Chaque browser a sa propre façon de travailler.

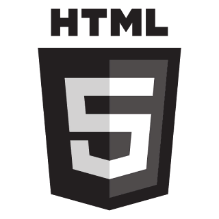


Figure 16 :logo html

**2.4. Une feuille de style: CSS :**

Les feuilles de style en cascade, généralement appelées CSS de l’anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C). Introduit au milieu des années 1990, CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années 2000.

****

Figure 17 : logo CSS

**2.5. JavaScript :**

JavaScript est un langage de scripts qui incorporé aux balises Html, permet d'améliorer la présentation et l'interactivité des pages Web. JavaScript est donc une extension du code Html des pages Web. Les scripts, qui s'ajoutent ici aux balises HTML, peuvent en quelque sorte être comparés aux macros d'un traitement de texte.

Ces scripts vont être gérés et exécutés par le browser lui-même sans devoir faire appel aux ressources du serveur. Ces instructions seront donc traitées en direct et surtout sans retard par le navigateur.

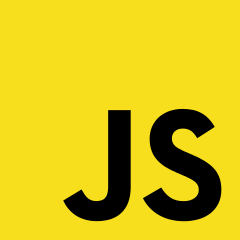


Figure 18 :logo JavaScript

**2.6. Bootstrap:**

Bootstrap est une collection d’outils utile à la création du design (graphisme, animation et interactions avec la page dans le navigateur etc.) de sites et d’applications web. C’est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C’est un Framework CSS et JavaScript développé par la société californienne Twitter.

****

Figure 19 : logo Bootstrap

﻿

**3. Les outils de développement :**

**3.1. XAMPP:**

XAMPP est un ensemble de logiciels permettant de mettre en place facilement un serveur Web, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres (**XA**pache**M**ySQL**P**erl**P**HP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus.



Figure 20 : logo XAMPP

﻿

**3.2. Mozilla Firefox:**

Mozilla Firefox est un navigateur Web, développé et distribué par la Mozilla Foundation aidée de centaines de bénévoles grâce aux méthodes de développement du logiciel libre/open source et à la liberté du code source.



Figure 21 : logo Firefox

**3.3. Power AMC Designer :**

PowerAMC désigner est un logiciel de modélisation. Il permet de modéliser les traitements informatiques et leurs bases de données associées. Créé par SDP sous le nom AMC\*Designor, racheté par Powersoft, ce logiciel est produit par Sybase depuis le rachat par cet éditeur en 1995.

****

Figure 22 : logo PowerAMC

**3.4. Visual Studio Code :**

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS3.Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code (IntelliSense4.), les snippets, la refactorisation du code et Git intégré. Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires.

Le code source de Visual Studio Code provient du projet logiciel libre et open source VS Code de Microsoft publié sous la licence MIT permissive, mais les binaires compilés constituent un freeware, c'est-à-dire un logiciel gratuit pour toute utilisation mais propriétaire.

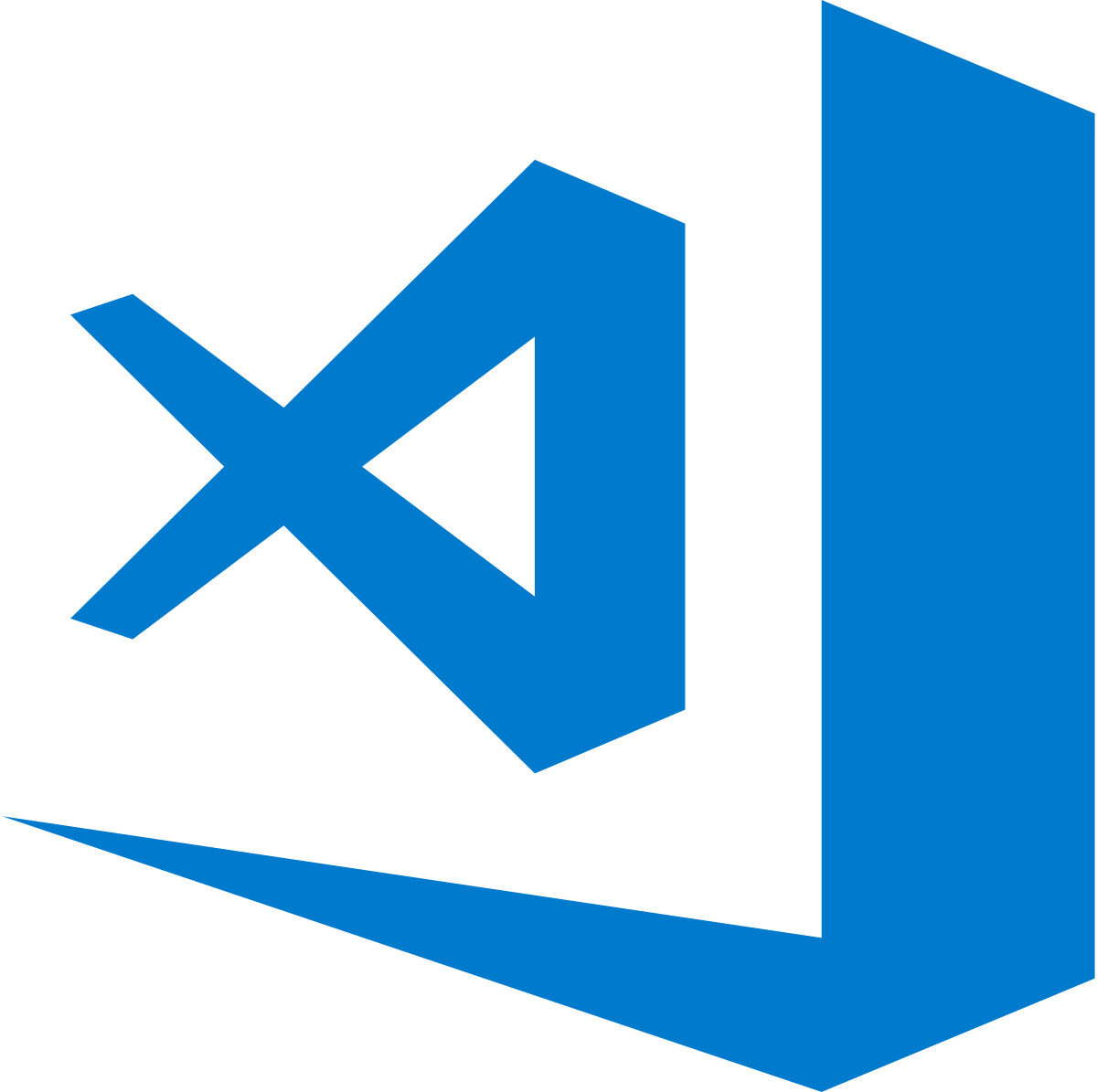


Figure 23 : logo Visual Studio Code

# CHAPITRE 4 : Réalisation du projet

﻿**1. Introduction :**

Dans cette partie nous traitons les différents détails liés à l'implémentation et au développement de l'application déjà conçue et modélisée dans la partie précédente. Nous nous allons illustrer par des imprimes écrans la description du travail réalisé, ainsi que l'environnement matériel et logiciel qui fera l'objet de la dernière partie de ce dernier chapitre.

﻿**2. Système de gestion de base de données :**

Pour implémenter notre base de données, nous avons utilisé phpmyAdmin comme SGBD (Système de gestion de base de données). PhpmyAdmin est un SGBD relationnel. C'est un logiciel libre. Il est caractérisé par sa rapidité et sa fiabilité. Il a été développé à l'origine pour gérer de très grandes bases de données.

﻿

**3. Travail réalisé :**

Après avoir mené à terme les différentes phases précitées du projet, on est dans la mesure de commencer la partie de réalisation pour chaque itération avec une vision claire et précise. Dans ce qui suit, on va présenter une description du site web réalisé via quelques interfaces.

﻿

**3.1. Coté Client (internaute):**

**3.1.1. Page d'Accueil :**

Devant la page d'accueil l'internaute a le choix de consulter les rubriques, voir les voitures de l'agence, voir les différents services présentés par le site, voir les voitures disponibles pour la réservation.



Figure 24 : Page d’accueil

﻿

**3.1.2. La phase de réservation d'une voiture par des illustrations :**

Lorsque le client visite la page d'accueil il remarque la présence d'un formulaire pour spécifier les critères de sa recherche (la date dont la quelle il veut effectuer la réservation et la classe de la voiture souhaitée), ce qui illustre la figure suivante

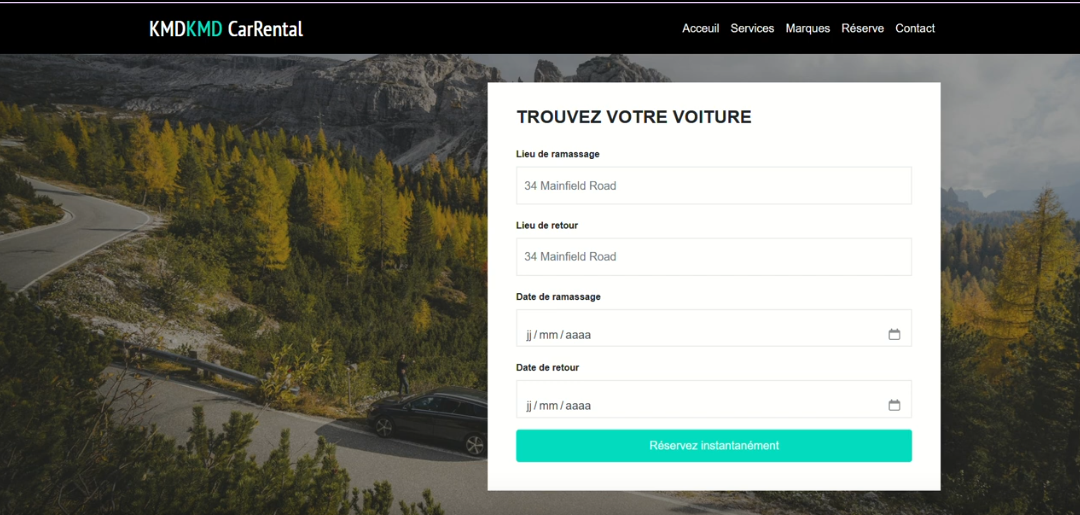


Figure 25 : Formulaire de réservation

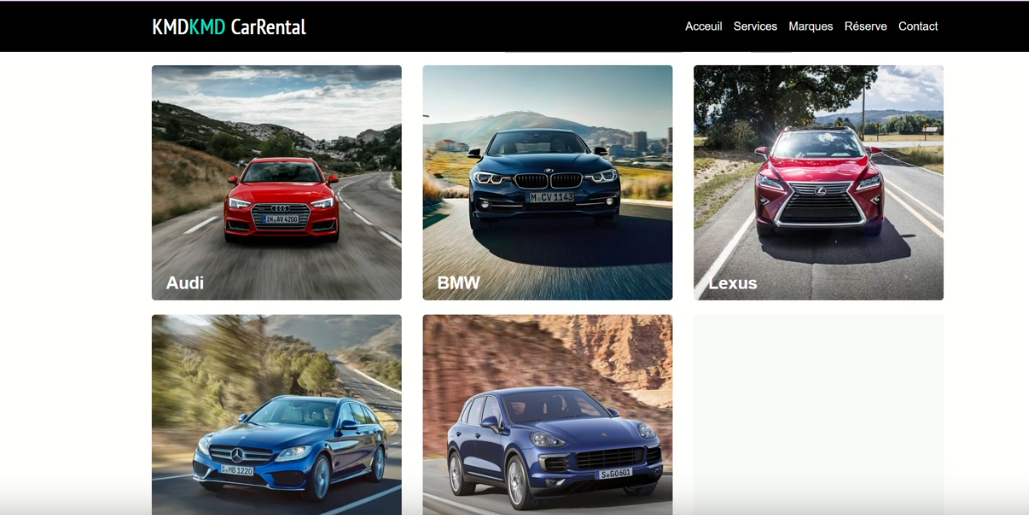


Figure 26 : Les voitures disponibles pour la réservation

Une image contenant texte, capture d’écran, jauge, multimédia

Description générée automatiquement

Figure 27 : page de réservation

Une image contenant texte, logiciel, Page web, Icône d’ordinateur

Description générée automatiquement

Figure 28 : Formulaire de contact

﻿

**3.2. Coté Administrateur (Webmaster) :**

**3.2.1. Authentification de l'administrateur :**

Pour pouvoir effectuer ses fonctionnalités l'administrateur doit s'authentifier en saisissant un Login et un mot de passe. Cette authentification est obligatoire pour passer en mode administration. Après vérification de login et de mot de passe, on passe à la page d'administration.

Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement

Figure 29 : formulaire d’authentifier administrateur

﻿

A partir de cette page l'administrateur peut gérer son site, il effectue toutes ses opérations :

* Gérer les voitures:
* Ajouter une voiture.
* Supprimer une voiture.
* Modifier les informations d'une voiture.
* Gérer les réservations:
* Ajouter une réservation.
* Annuler une réservation.
* Modifier les informations d'une réservation

✓ Gérer les clients :

* Ajouter un client.
* Supprimer un client.
* Modifier les informations d'un client.

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

Figure 30 : Gestion des clients

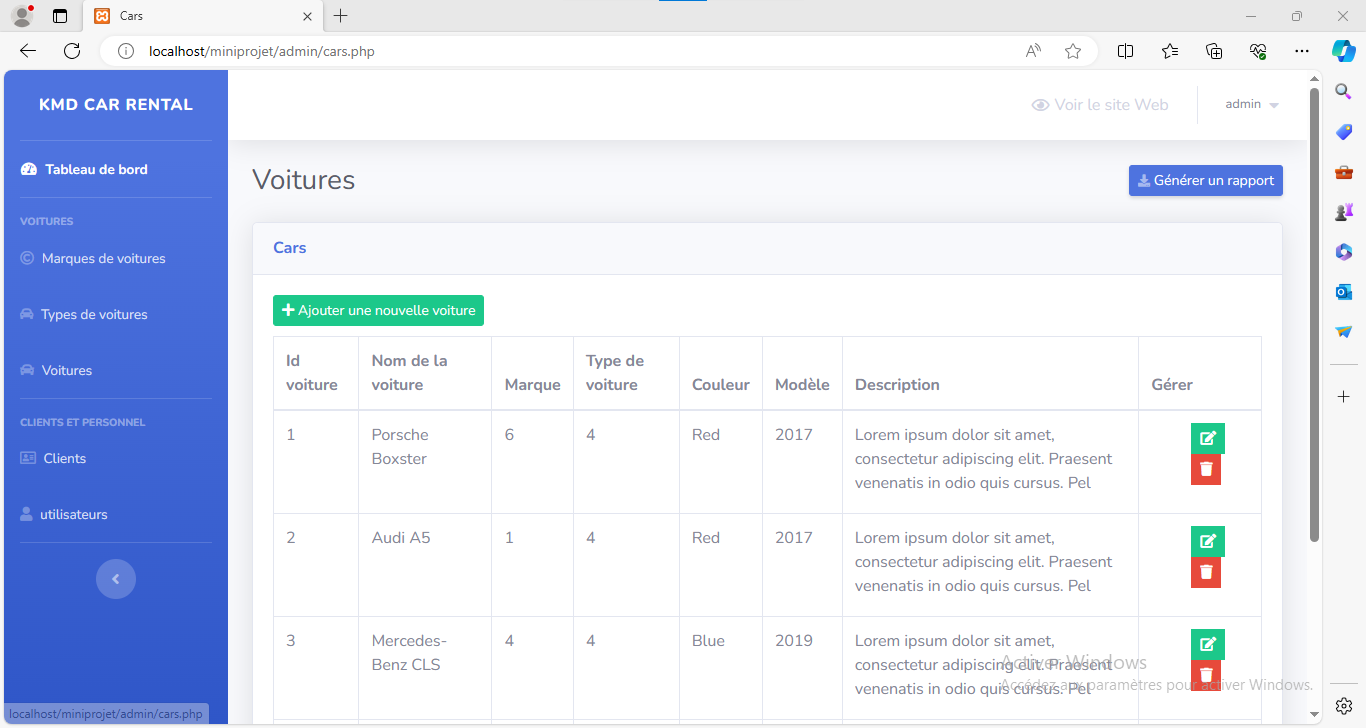


Figure 31 : Gestion des voitures

# 

Figure 32 :gestion des types de voitures

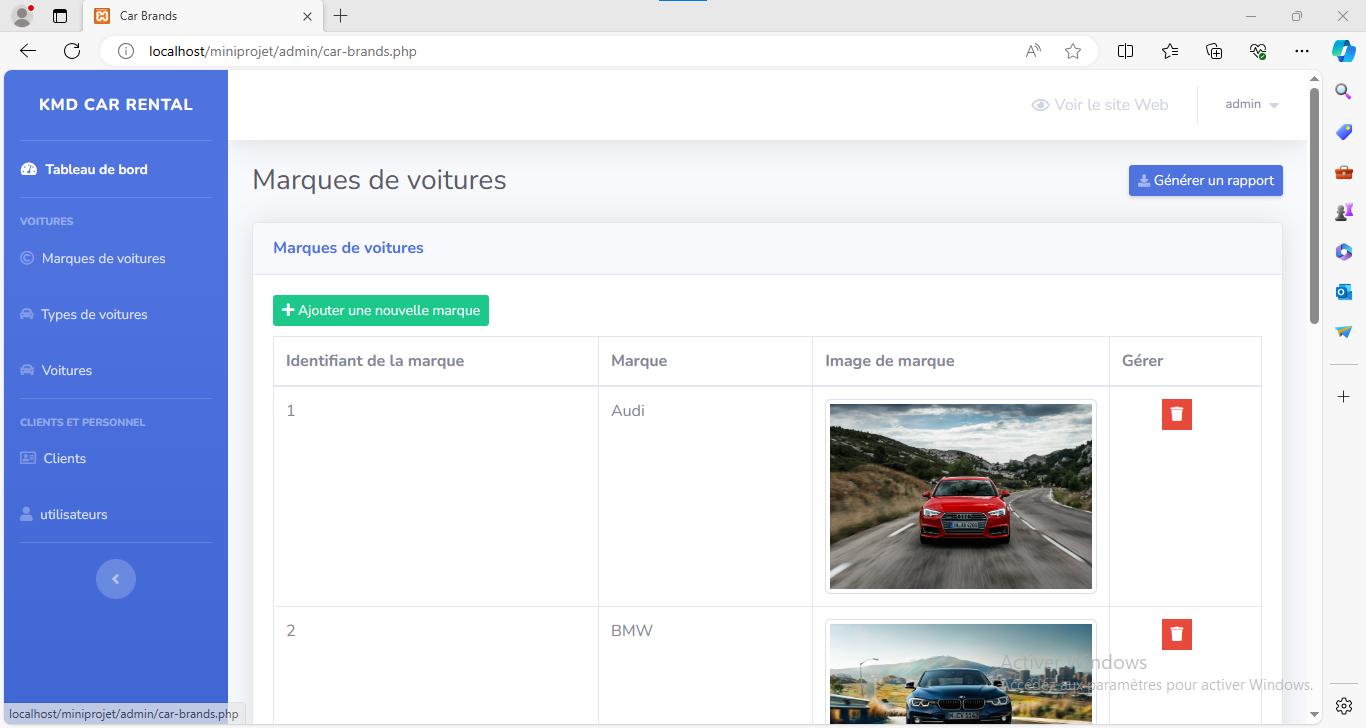
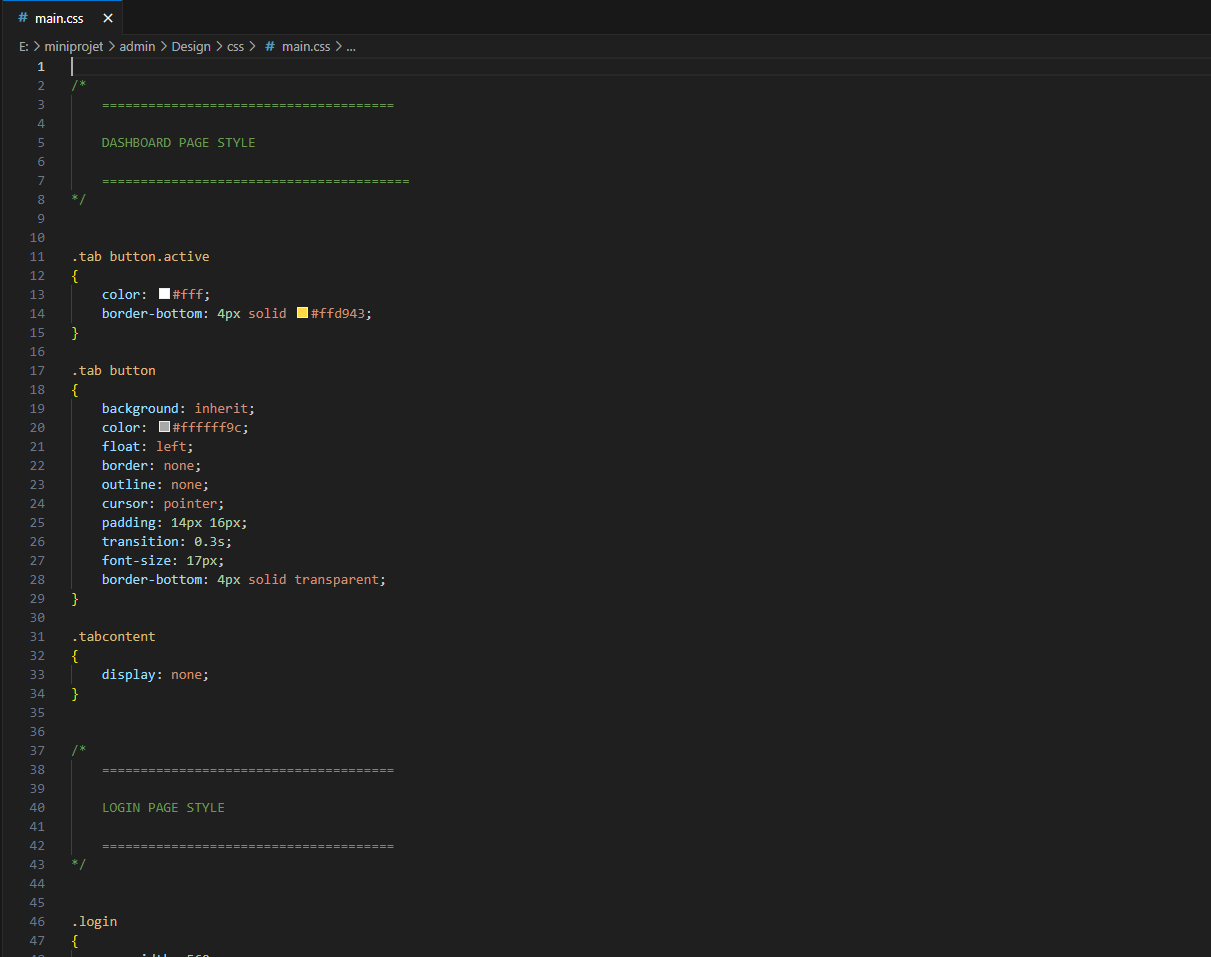


Figure 33 : Gestion de marques des voitures

# ***main.css***



* **Introduction**
  + Présentation rapide du fichier CSS main.css
  + Contexte général du style visuel du site web
* **Styles de l'Espace d'Administration (Dashboard)**
  + Onglets de navigation pour la page d'administration
  + Mise en évidence visuelle de l'onglet actif
  + Utilisation de la couleur et des bordures pour améliorer la convivialité
* **Styles de la Page de Connexion (Login)**
  + Design du formulaire de connexion
  + Utilisation d'une boîte d'ombre pour accentuer le formulaire
  + Responsivité de la mise en page pour différentes tailles d'écran
* **Styles des Éléments d'Entrée (Input)**
  + Style uniforme pour les champs de texte et mots de passe
  + Effets visuels lors de la saisie et de la mise au point des champs
  + Boutons de soumission avec des couleurs attractives et effets de survol
* **Gestion de la Sécurité**
  + Utilisation de couleurs pour indiquer l'état actif
  + Styles spécifiques pour les boutons de soumission sécurisés
* **Typographie**
  + Taille, couleur et style des polices
  + Utilisation de majuscules pour les titres et les étiquettes
* **Boîte à Outils pour la Réactivité**
  + Utilisation de médias requêtes pour assurer la réactivité
  + Ajustements de mise en page pour les écrans de petite taille
* **Conclusion**
  + Récapitulation des points clés du fichier CSS main.css
  + Possibilités d'améliorations futures

# 

# 

# 

# 

# **la Fonction JavaScript de la Page de Tableau de Bord**



* **Introduction**
  + Présentation de la fonction openTab dans le contexte de la page de tableau de bord
  + Objectif : Gérer l'affichage des onglets et des contenus associés
* **Structure de la Fonction**
  + Paramètres de la fonction : evt (événement), tabName (nom de l'onglet)
  + Utilisation de variables locales : i, tabcontent, tablinks
* **Fonctionnement de la Fonction**
  + Masquage de tous les contenus d'onglets
  + Suppression de la classe "active" de tous les liens d'onglets
  + Affichage du contenu de l'onglet sélectionné
  + Ajout de la classe "active" au lien d'onglet sélectionné
* **Gestion des Événements**
  + La fonction est déclenchée par un événement, généralement un clic sur un lien d'onglet
  + Utilisation de evt pour identifier l'onglet à afficher
* **Utilisation de la Fonction**
  + Présentation des éléments HTML liés à la fonction (onglets et contenus associés)
  + Exemple d'appel de la fonction dans le contexte de la page de tableau de bord
* **Conclusion**
  + Récapitulation du rôle de la fonction openTab
  + Importance de cette fonction dans la gestion des onglets de la page de tableau de bord

# 

# 

# 

# ***car-brands.php***

## 

**Introduction**

* + Objectif de la page : Gestion des marques de voitures
  + Utilisation de PHP pour la gestion de session et inclusion de fichiers
  + Définition dynamique du titre de la page avec PHP
* **Structure HTML**
  + Utilisation de Bootstrap pour la mise en page
  + Barre de navigation latérale avec des liens vers d'autres sections du site
  + Barre supérieure avec des liens vers le site public et des options utilisateur
* **Section Principale**
  + Gestion des sessions : Vérification de l'authentification de l'utilisateur
  + Contenu principal divisé en différentes sections (ajout, suppression de marques)
  + Affichage d'alertes en cas d'ajout ou de suppression réussi
* **Formulaire d'Ajout de Nouvelle Marque**
  + Utilisation de Vue.js pour la validation côté client
  + Saisie du nom de la marque et téléchargement d'une image
  + Prévisualisation de l'image sélectionnée
  + Gestion des erreurs avec affichage dynamique
* **Tableau des Marques de Voitures**
  + Affichage des marques actuellement enregistrées dans une table
  + Possibilité de supprimer une marque avec confirmation
* **Modalités pour la Suppression**
  + Utilisation de modals Bootstrap pour confirmer la suppression
  + Model avec le nom de la marque pour confirmation
* **Conclusion**
  + Résumé des fonctionnalités de la page "car-brands.php"
  + Possibilités d'améliorations futures

# ***car-types.php***

## 

* **Introduction**
  + Objectif de la page : Gestion des types de voitures
  + Utilisation de PHP pour la gestion de session et inclusion de fichiers
  + Définition dynamique du titre de la page avec PHP
* **Structure HTML**
  + Utilisation de Bootstrap pour la mise en page
  + Barre de navigation latérale avec des liens vers d'autres sections du site
  + Barre supérieure avec des liens vers le site public et des options utilisateur
* **Section Principale**
  + Gestion des sessions : Vérification de l'authentification de l'utilisateur
  + Contenu principal divisé en différentes sections (ajout, suppression de types)
  + Affichage d'alertes en cas d'ajout ou de suppression réussi
* **Formulaire d'Ajout de Nouveau Type**
  + Saisie du libellé et de la description du type de voiture
  + Utilisation de Vue.js pour la validation côté client
  + Gestion des erreurs avec affichage dynamique
* **Tableau des Types de Voitures**
  + Affichage des types actuellement enregistrés dans une table
  + Possibilité de supprimer un type avec confirmation
* **Modalités pour la Suppression**
  + Utilisation de modals Bootstrap pour confirmer la suppression
  + Modal avec le libellé du type pour confirmation
* **Conclusion**
  + Résumé des fonctionnalités de la page "car-types.php"
  + Possibilités d'améliorations futures

# 

# ***cars.php***

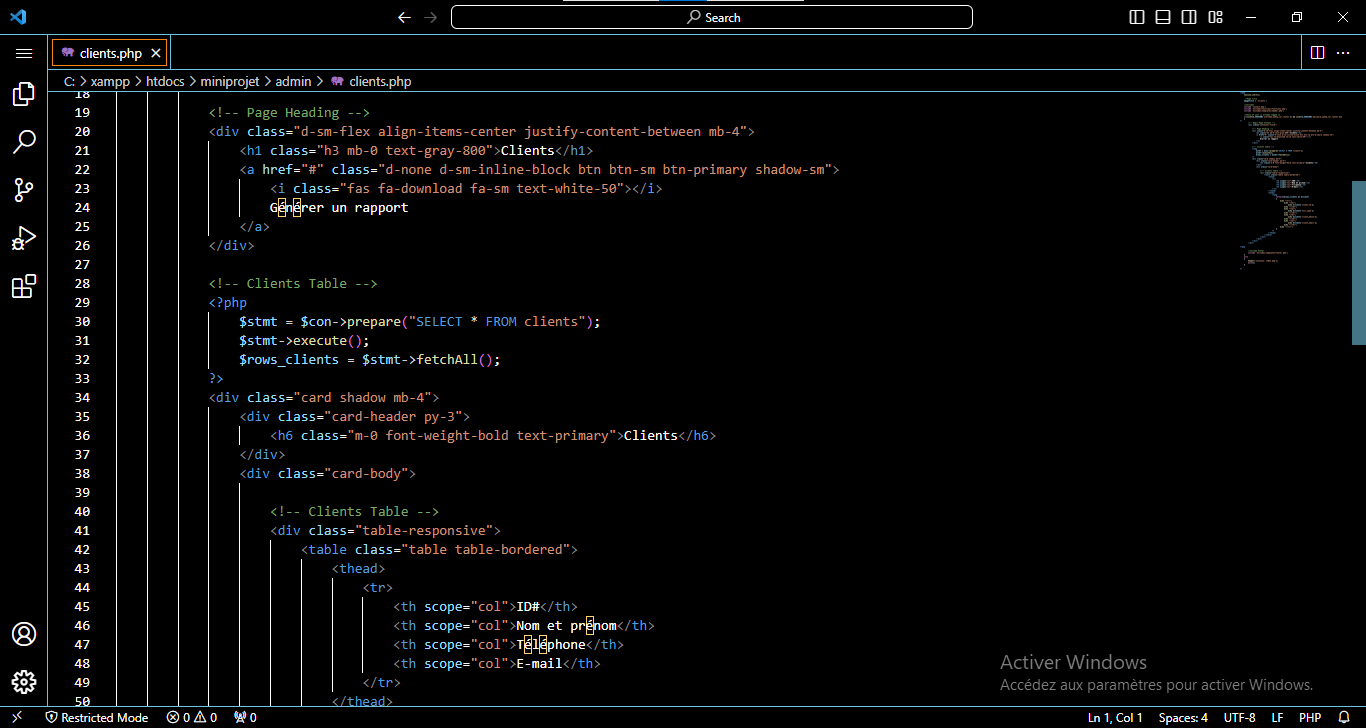


## 

* **Introduction**
  + Objectif de la page : Gestion des voitures
  + Utilisation de PHP pour la gestion de session et inclusion de fichiers
  + Définition dynamique du titre de la page avec PHP
* **Structure HTML**
  + Utilisation de Bootstrap pour la mise en page
  + Barre de navigation latérale avec des liens vers d'autres sections du site
  + Barre supérieure avec des liens vers le site public et des options utilisateur
* **Section Principale**
  + Gestion des sessions : Vérification de l'authentification de l'utilisateur
  + Contenu principal divisé en différentes sections (ajout de voiture, tableau de voitures)
  + Affichage d'alertes en cas d'ajout réussi ou d'erreurs
* **Formulaire d'Ajout de Nouvelle Voiture**
  + Sélection de la marque et du type de voiture à partir des données existantes
  + Saisie de la couleur, du modèle et de la description
  + Utilisation de Vue.js pour la validation côté client
  + Gestion des erreurs avec affichage dynamique
* **Tableau des Voitures**
  + Affichage des voitures actuellement enregistrées dans une table
  + Possibilité de supprimer une voiture avec confirmation
  + Bouton d'édition (non fonctionnel dans l'état actuel)
* **Modalités pour la Suppression**
  + Utilisation de modals Bootstrap pour confirmer la suppression
  + Modal avec une option de confirmation générique
* **Conclusion**
  + Résumé des fonctionnalités de la page "cars.php"
  + Possibilités d'améliorations futures

# 

# ***clients.php***



* **Introduction**
  + Objectif de la page : Affichage des clients enregistrés
  + Utilisation de PHP pour la gestion de session et inclusion de fichiers
  + Définition dynamique du titre de la page avec PHP
* **Structure HTML**
  + Utilisation de Bootstrap pour la mise en page
  + Barre de navigation latérale avec des liens vers d'autres sections du site
  + Barre supérieure avec des liens vers le site public et des options utilisateur
* **Section Principale**
  + Gestion des sessions : Vérification de l'authentification de l'utilisateur
  + Affichage de la liste des clients enregistrés
* **Tableau des Clients**
  + Affichage des clients actuellement enregistrés dans une table
  + Colonnes : ID, Nom complet, Numéro de téléphone, E-mail
  + Possibilité d'ajouter d'autres fonctionnalités au tableau si nécessaire
* **Conclusion**
  + Résumé des fonctionnalités de la page "clients.php"
  + Possibilités d'améliorations futures

***connect.php***

******

un script PHP permettant d'établir une connexion à une base de données MySQL en utilisant l'extension PDO (PHP Data Objects). Voici une explication du code :

* **Détails de la Connexion :**
  + $dsn : Nom de la source de données, spécifie le pilote de base de données, l'hôte et le nom de la base de données.
  + $user : Nom d'utilisateur de la base de données.
  + $pass : Mot de passe de la base de données.
  + $option : Un tableau contenant des options supplémentaires. Dans ce cas, il définit le jeu de caractères sur UTF-8.
* **Bloc Try-Catch :**
  + Le bloc try tente de créer une nouvelle instance PDO en utilisant les détails de connexion fournis.
  + En cas de succès, il définit le mode d'erreur pour déclencher des exceptions (PDO::ERRMODE\_EXCEPTION).
  + S'il y a une exception lors de la tentative de connexion, le bloc catch est exécuté.
  + Dans le bloc catch, un message d'erreur est affiché, indiquant que la connexion à la base de données a échoué. Le message d'erreur réel est obtenu à partir de l'exception ($ex->getMessage()).
* **Commentaires :**
  + Il y a une ligne //echo 'Good Very Good !'; mise en commentaire, qui semble être un espace réservé ou une instruction de débogage. Elle n'est actuellement pas active.
* **Conclusion :**
  + Ce script est une façon standard de se connecter à une base de données MySQL en utilisant PDO en PHP.
  + Il est bon de gérer les erreurs de connexion à la base de données de manière élégante, comme le montre le bloc try-catch.

*Dashboard*

**

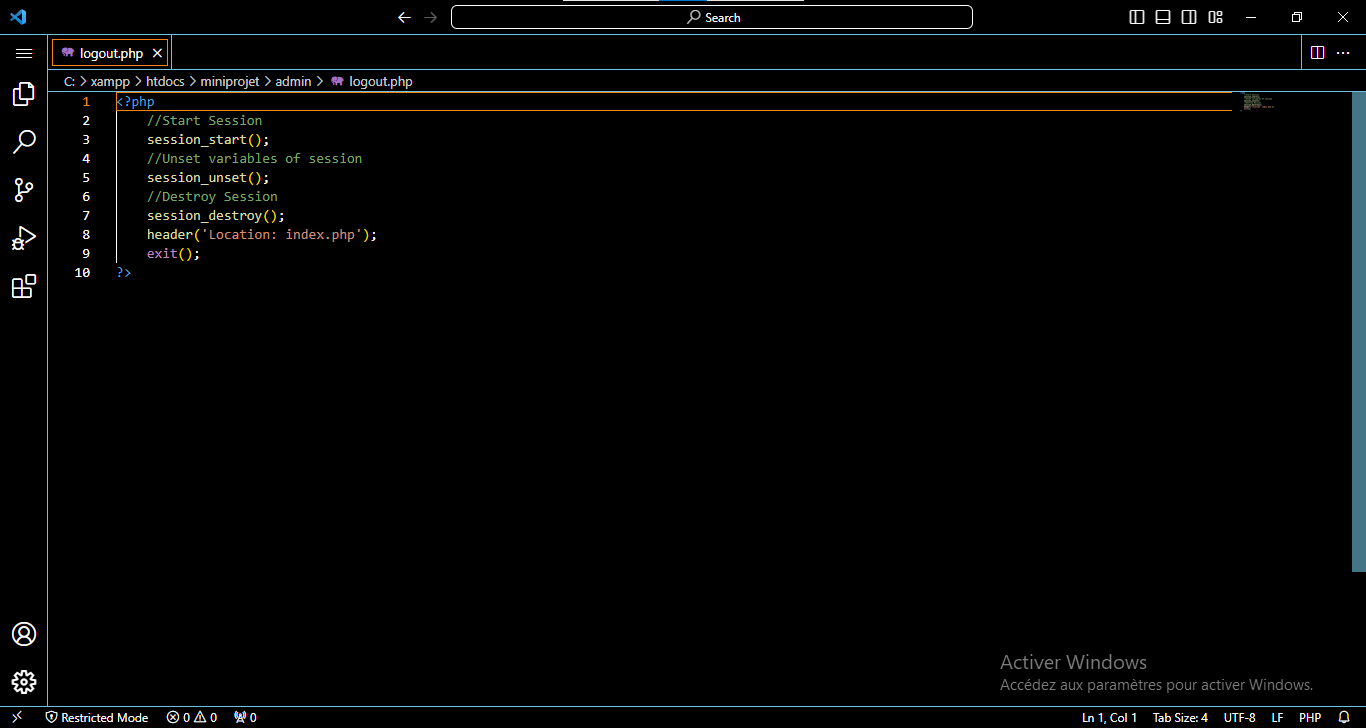
* **Connexion à la Base de Données :**
  + Utilisation de PDO pour établir une connexion à la base de données MySQL dans le fichier connect.php.
* **Vérification de la Session Utilisateur :**
  + Utilisation de session\_start() pour gérer les sessions.
  + Vérification si l'utilisateur est connecté en vérifiant la présence de variables de session ($\_SESSION['username\_yahya\_car\_rental'] et $\_SESSION['password\_yahya\_car\_rental']).
* Affichage du Tableau de Bord :
  + Affichage de statistiques telles que le nombre total de clients, marques de voitures, voitures et réservations à venir.
  + Utilisation de la fonction countItems pour compter le nombre d'éléments dans différentes tables.
* **Annulation de Réservation :**
  + Possibilité d'annuler une réservation avec la raison d'annulation spécifiée.
  + Utilisation d'un formulaire et d'une requête SQL pour mettre à jour l'état de la réservation.
* **Affichage des Réservations :**
  + Affichage des réservations à venir, de toutes les réservations et des réservations annulées dans des onglets distincts.
  + Utilisation de modaux pour confirmer l'annulation d'une réservation.
* Utilisation de l'API Tab (Javascript) :
  + L'utilisation de la fonction openTab pour basculer entre les onglets du tableau de bord.
* **Mise en Page Responsive :**
  + Utilisation des classes Bootstrap pour créer une mise en page réactive.
* **Gestion des Erreurs :**
  + Utilisation de blocs try et catch pour gérer les erreurs liées à la base de données.
* **Redirection si Non Connecté :**
  + Si l'utilisateur n'est pas connecté, il est redirigé vers la page d'accueil (index.php).

*index.php*

**

* Vérification de la Session Utilisateur :
  + Si l'utilisateur est déjà connecté, il est redirigé vers le tableau de bord (dashboard.php).
* **Affichage de la Page de Connexion :**
  + Utilisation de PHP pour inclure des fichiers (connect.php et functions.php).
  + Utilisation de Bootstrap pour la mise en page et le style.
* **Formulaire de Connexion :**
  + Utilisation d'un formulaire HTML avec des champs pour le nom d'utilisateur (username) et le mot de passe (password).
  + Utilisation de la méthode POST pour soumettre le formulaire.
* **Traitement du Formulaire :**
  + Vérification si le formulaire a été soumis ($\_SERVER["REQUEST\_METHOD"] == "POST").
  + Extraction des valeurs du formulaire ($username et $password).
  + Utilisation de PDO pour vérifier l'existence de l'utilisateur dans la base de données.
* **Redirection Réussie :**
  + Si l'utilisateur est trouvé dans la base de données, les informations de session sont définies et l'utilisateur est redirigé vers le tableau de bord (dashboard.php).
* **Gestion des Erreurs :**
  + Affichage d'une alerte rouge si les informations de connexion sont incorrectes.
* Liens Supplémentaires :
  + Un lien "Forgot your password?" pointant vers une éventuelle réinitialisation du mot de passe.
* **Pied de Page :**
  + Affichage d'un pied de page avec le message de copyright.
* Inclusion de Scripts JavaScript :
  + Inclusion des bibliothèques JavaScript comme jQuery, sweetalert, Bootstrap, etc.

*Logout.php*

**

* **Démarrer la Session :**
  + session\_start() est utilisé pour démarrer la session.
* **Supprimer les Variables de Session :**
  + session\_unset() est utilisé pour supprimer toutes les variables de session.
* **Détruire la Session :**
  + session\_destroy() est utilisé pour détruire complètement la session.
* **Redirection vers la Page de Connexion :**
  + Une fois la session détruite, l'utilisateur est redirigé vers la page de connexion (index.php).
* **Terminer le Script :**
  + exit() est utilisé pour s'assurer que le script se termine après la redirection.

*users.php*

**

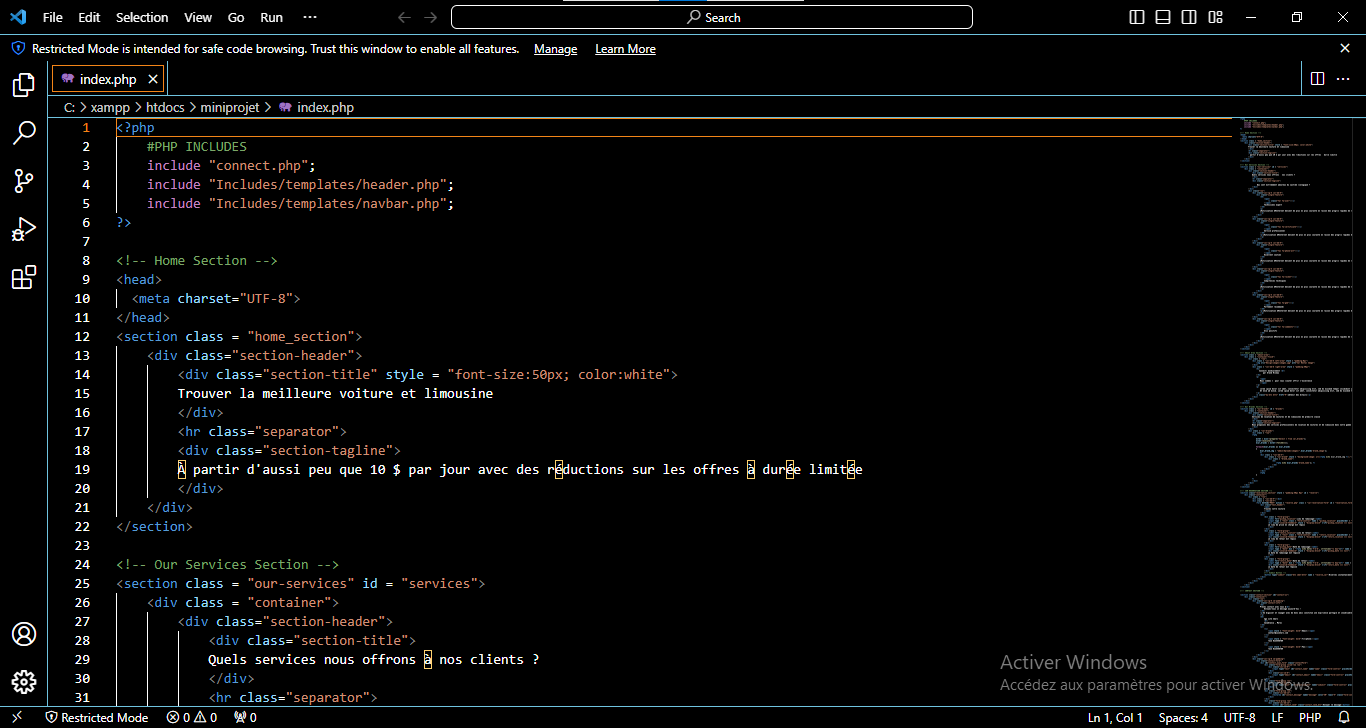
* **Démarrage de la Session :**
  + session\_start() : Démarre ou reprend la session.
* **Titre de la Page et Inclusions :**
  + pageTitle est défini sur 'Users'.
  + Des fichiers importants sont inclus, tels que 'connect.php', 'functions.php', et 'header.php'.
* **Vérification de la Connexion de l'Utilisateur :**
  + Vérifie si l'utilisateur est déjà connecté en vérifiant l'existence des variables de session username\_yahya\_car\_rental et password\_yahya\_car\_rental.
* **Contenu de la Page :**
  + Si l'utilisateur est connecté, le contenu de la page est affiché.
  + Le titre de la page, un en-tête avec un lien de téléchargement factice, et un tableau de données utilisateur extrait de la base de données sont affichés.
* **Récupération des Utilisateurs :**
  + Une requête SQL est utilisée pour récupérer tous les utilisateurs de la base de données.
  + Les résultats sont ensuite affichés dans un tableau.
* **Affichage des Données Utilisateur :**
  + Une boucle foreach est utilisée pour parcourir les utilisateurs et afficher leurs informations dans les cellules du tableau.
* **Inclusion du Pied de Page :**
  + Le pied de page est inclus à la fin.
* Redirection si Non Connecté :
  + Si l'utilisateur n'est pas connecté, il est redirigé vers la page d'index.

*Car rental.sql*

**

* **Table cars :**
  + Stocke les informations sur les voitures, telles que le nom, la marque, le type, la couleur, le modèle et la description.
* **Table car\_brands :**
  + Contient des informations sur les marques de voitures, y compris le nom de la marque et le chemin de l'image de la marque.
* **Table car\_types :**
  + Décrit différents types de voitures (Sedan, Coupe, HATCHBACK) avec des libellés et des descriptions associées.
* **Table clients :**
  + Enregistre les informations sur les clients, tels que l'ID client, le nom complet, l'e-mail et le numéro de téléphone.
* **Table reservations :**
  + Stocke les détails des réservations, y compris l'ID de réservation, l'ID du client, l'ID de la voiture, la date et le lieu de prise en charge, la date et le lieu de restitution, le statut d'annulation et la raison d'annulation.
* **Table users :**
  + Contient les informations sur les utilisateurs de l'application, y compris l'ID utilisateur, le nom d'utilisateur, l'e-mail, le nom complet, le mot de passe et l'ID de groupe.

***index.php***

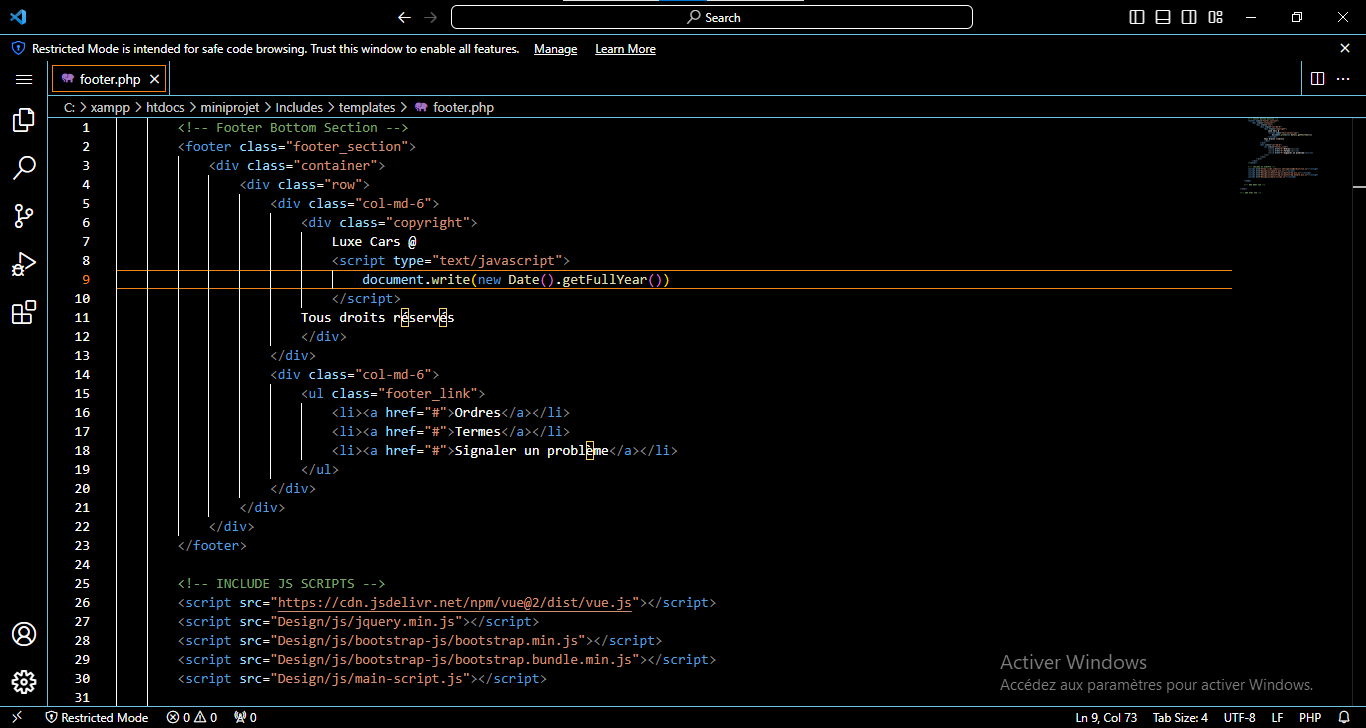
******

* **Section Home :**
  + Affiche une section d'accueil avec un titre, une sous-titre et une hr.
* **Section Our Services :**
  + Présente les services offerts aux clients avec trois caractéristiques affichées sous forme de cartes.
* **Section About Area :**
  + Divisée en deux colonnes, avec une colonne contenant une image et l'autre contenant un titre, un sous-titre, un paragraphe et un lien.
* Section Our Brands :
  + Affiche les marques de voitures à travers des images en utilisant les données récupérées à partir de la table car\_brands de la base de données.
* **Section Car Reservation :**
  + Affiche un formulaire de réservation de voiture. Les données sont validées côté client à l'aide de Vue.js avant de soumettre le formulaire.
* **Section Contact :**
  + Affiche les informations de contact et un formulaire de contact. Cependant, le formulaire ne semble pas être entièrement implémenté en PHP/Vue.js. Il utilise une classe contactForm qui pourrait nécessiter une implémentation supplémentaire pour gérer la soumission et la validation côté serveur.
* **Section Widget (Footer) :**
  + Affiche les informations de contact, des icônes de médias sociaux et un formulaire d'abonnement à la newsletter.
* **Script Vue.js :**
  + Utilise Vue.js pour la validation du formulaire de réservation de voiture. Si les champs requis ne sont pas remplis, il les met à null pour déclencher l'affichage des messages d'erreur. La validation côté serveur (reserve.php) est recommandée pour des raisons de sécurité.

*reserve.php*

**

* Démarrage de session et inclusion des fichiers :
  + Démarrez la session et incluer les fichiers nécessaires, tels que connect.php, header.php, navbar.php, et functions.php.
* Affichage de la section de bannière pour la réservation :
  + Une section est affichée avec le titre "Réservez votre voiture".
* **Section de réservation de voiture :**
  + Si le formulaire de réservation initial a été soumis, on stocke les informations dans la session.
  + Si le formulaire de réservation final est soumis, on récupère les données de la session, puis on effectue des opérations de réservation et d'insertion dans la base de données.
  + On utilise la méthode POST et les données sont traitées côté serveur.
  + On affiche ensuite le formulaire de réservation final avec les informations de réservation et une liste de voitures disponibles.
* Vue.js pour la validation du formulaire :
  + On utilise Vue.js pour la validation du formulaire final. Si les champs requis ne sont pas remplis, on les met à null pour déclencher l'affichage des messages d'erreur.
* **Affichage de messages d'erreur ou de succès :**
  + On affiche des messages d'erreur ou de succès selon le résultat de l'opération de réservation.
* *footer.php*

**

* + inclure le fichier footer.php à la fin.

**5. Conclusion :**

Ce chapitre a pour objectif de présenter les différents aspects pratiques de l'application.

Malgré les difficultés lors de langage de programmation PHP et de la base de données qu'on a utilisé dans notre application, nous avons pu finalement modéliser la partie conceptuelle de notre site. A cette étape la réalisation du projet devient claire suivant la démarche que nous avons suivie.

# CONCLUSION GENERALE

Au cours de ce projet, nous avons étudié un secteur d'activité très intéressant : c'est celui du service de location des voitures de luxe en ligne.

L'objectif escompté par ce présent travail est de concevoir, de développer un site web pour une agence de location des voitures de luxe en ligne.

En achevant ce rapport, il est indispensable de signaler les aspects bénéfiques de ce projet :

✓ Sur le plan méthodologique, il a permis de mieux maîtriser le langage de modélisation UML.

✓ Sur le plan pratique, nous avons été confrontés à la gestion d'un projet depuis la phase d'analyse jusqu'à l'implémentation en passant par la conception et la réalisation.

Il nous a permis de faire de nouveaux contacts avec plusieurs personnes s'impliquant de près ou de loin aux nouvelles technologies et de côtoyer le monde professionnel.

# Bibliographie

[https://fr.wikipedia.org](https://fr.wikipedia.org/wiki/Wikip%C3%A9dia:Accueil_principal)

<https://openclassrooms.com/fr/>

<https://www.w3schools.com/>

<https://stackoverflow.com/>