

Table des matières

Préambule.....	2
INTRODUCTION DU PROJET.....	2
LIVRABLES.....	3
Fichiers de création et d'alimentation de la base de donnée de l'appli & Documentation technique & Manuel processus d'installation à l'intention des correcteurs.....	3
1. Formulaire de Demande d'Information.....	5
2. Gestion Administrateur et Employés.....	6
3. Mise à Jour du Catalogue de Véhicules.....	7
4. Gestion des Comptes Utilisateurs et Fonctionnalités Administratives.....	9
Comptes Administrateur et Employés.....	9
Fonctionnalités de Gestion des Véhicules.....	9
Sécurité et Accès.....	9
Objectif du Système de Gestion.....	10
5. Configuration de l'environnement de travail.....	10
Liste des logiciels nécessaires.....	10
Langage et frameworks.....	11
Environnement Front-end.....	11
Environnement Back-end.....	12
Installation et configuration.....	12
Environnement de Production.....	13
Arborescence des fichiers.....	13
Configuration Locale avec WAMP64.....	14
Gestion de la Base de Données MySQL.....	14
Guide de Développement.....	14
Architecture.....	15
Gestion des Dépendances.....	15
Sécurité.....	15
Guide de Déploiement (pas encore déployé).....	16
Tests et Maintenance.....	16
Manuel du projet.....	16
6. Mise en œuvre.....	16
1. Choix des Technologies.....	16
2. Diagrammes et Conception.....	16
3. Charte Graphique.....	17
4. Manuel d'Installation et de Test.....	17
Installation en Local.....	17

Préambule

À l'intention de l'ensemble des correcteurs,
Résumé simple Sandrine Blandamour aspirante développeuse full stack :

Bonjour,

Je m'appelle Sandrine, et je suis passionnée par le développement web. J'ai choisi cette formation ; graduate développeuse Front-end, car elle allie technologie, créativité et constante évolution. Ma curiosité naturelle m'a poussée à explorer de nombreux domaines, notamment la géopolitique, les marchés boursiers, les ETF, ainsi que l'intelligence artificielle et la technologie de l'information. Ces intérêts divers m'ont aidée à adopter une perspective globale dans mes projets de développement.

À côté de ma vie professionnelle, je m'adonne à plusieurs sports, dont l'équitation, le tennis, la plongée sous-marine, et la danse ce qui me permet d'avoir un équilibre entre mon esprit et mon corps.

En ce qui concerne mes aspirations futures, après avoir consolidé mes compétences en développement web auprès de développeurs passionnés, en entreprise ou en créant ma propre entreprise, je souhaite ajouter d'autres langages à mon panel de compétence, et également ce qui touche à la cybersécurité, un domaine que je considère primordial.

Aujourd'hui, je suis ici pour vous présenter mon projet ; un site vitrine pour le garage V.Parrot et je suis impatiente de recevoir vos retours.

INTRODUCTION DU PROJET

Vue d'ensemble du projet et réflexion :

Le projet consiste en un site vitrine pour un garage spécialisé en réparation automobile et en vente de voitures d'occasion. Le site offre plusieurs fonctionnalités clés pour améliorer l'expérience client et simplifier la gestion du garage :

Catalogue de Voitures d'Occasion

Affichage Dynamique des Véhicules : Les utilisateurs peuvent consulter les voitures disponibles à la vente, avec des informations détaillées affichées sur le site.

Le site intègre une base de données contenant une liste de marques de voitures avec des informations détaillées sur chaque véhicule (modèle, année, kilométrage, prix, etc.).

LIVRABLES

Fichiers de création et d'alimentation de la base de donnée de l'appli & Documentation technique & Manuel processus d'installation à l'intention des correcteurs

Pour une installation en local, il sera nécessaire d'installer le produit :

MYSQL 8.0

PHP 8.0.26

Apache 2.4.54

Au sein de PHP, les extensions suivantes devront à minima être activé :

extension=pdo_mysql

extension=openssl

la première étant pour l'interaction avec la base de données

La deuxième étant pour la partie échange de mail sur le protocole SSL/TLS.

Ensuite il faut que vous configuriez Apache pour qu'il puisse trouver les programmes sur le disque dur, en relation avec l'url/path saisie coté Front.

Pour cela modifier le fichier httpd.conf,

ajouter le listen 80 ou 443 en fonction du choix HTTP / HTTPS

et si HTTPS alors ajouter les balises de prises en comptes des certificats de l'hébergeur à savoir le fichier .cer certificats et le fichier de clé privé .key

Il faut aussi créer le lien entre le nom domaine/path pour le mettre en relation avec le projet sur le disque à savoir :

```
Alias /(1) "(2)/web/"
```

```
<Directory "(2)/web/">
```

```
    Options +Indexes +FollowSymLinks +MultiViews
```

```
    AllowOverride all
```

```
    Require local
```

```
</Directory>
```

ou (1), est le nom du path associé au domaine

et (2) est le nom du répertoire du projet.

Pour une mise en place plus simple car d'autres directives sont à mettre au sein d'apache notamment les directives permettant d'interpréter l'extention .php demandé via le client web pour que le processus PHP executable soit lancé, il est souhaitable d'installer un ensemble via WAMP sinon, il faut faire les choses à la main et à minima ajouter les lignes suivantes dans httpd.conf :

```
PHPIniDir "<repertoire d'installation apache>/bin"
```

```
LoadModule php_module "<repertoire d'installation php>/php8apache2_4.dll"
```

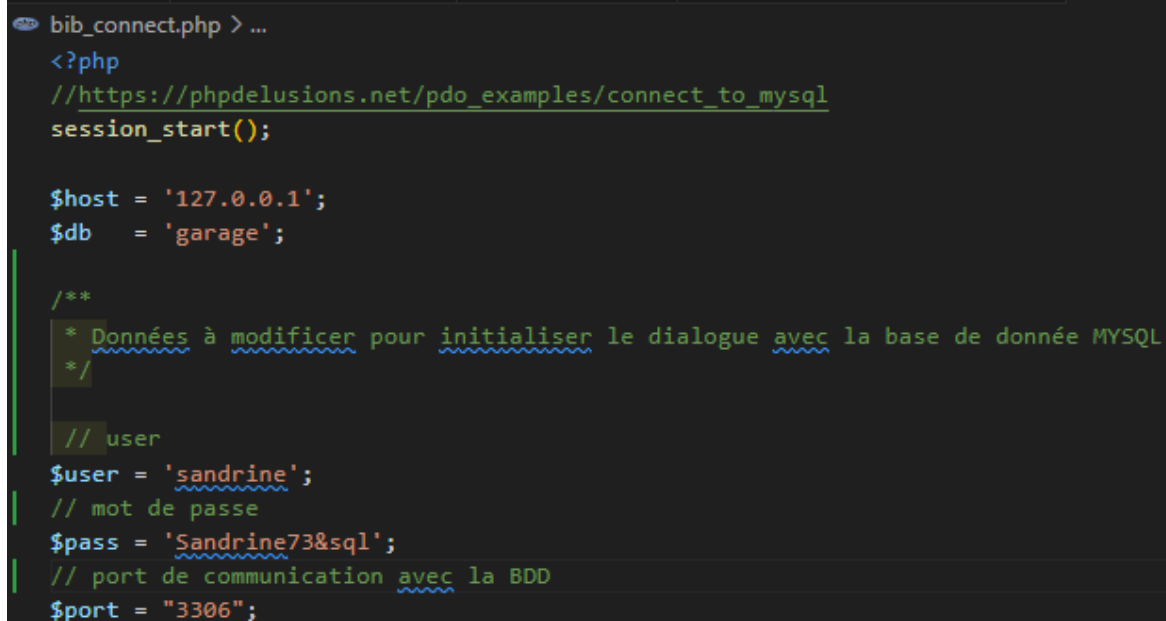
```
<IfModule dir_module>
  DirectoryIndex index.php index.php3 index.html index.htm
</IfModule>
```

```
# If the AddEncoding directives above are commented-out, then you
# probably should define those extensions to indicate media types:
AddType application/x-httpd-php .php
AddType application/x-httpd-php3 .php3
```

Au sein du projet, un répertoire nommé INIT, contient un fichier nommé DB_GARAGE.sql qui est à chargé et qui contient quelques Utilisateur et Vehicules pour « test » que vous pourrez supprimer à souhait.

Parr contre, comme vous allez mettre en place un serveur Mysql, donc avoir mis un login / mot de passe à la base de donnée, qui en fonction des version Mysql poeut être sur des ports différents exemple 3306 ou 3308 ou votre choix, il faut donc à minima aller modifier le fichier

bib/lib_connect.php pour indiquer ces valeurs à savoir :



```
bib_connect.php > ...
<?php
//https://phpdelusions.net/pdo_examples/connect_to_mysql
session_start();

$host = '127.0.0.1';
$db   = 'garage';

/**
 * Données à modifier pour initialiser le dialogue avec la base de donnée MYSQL
 */

// user
$user = 'sandrine';
// mot de passe
$pass = 'Sandrine73&sql';
// port de communication avec la BDD
$port = "3306";
```

En dernier, l'application envoi des mails, donc utilise un serveur SMTP qui doit donc être configure, ainsi le programme intégrant ces paramètres doit être modifié à savoir :

bib_mail.php

```

public static function getMail(){
    /**
     * A voir dans le futur ce qu'on appelle SPF et DKIM
     */
    $mail = new PHPMailer(true);
    $mail->isSMTP();
    $mail->CharSet = "UTF-8";
    $mail->Host = 'smtp.laposte.net';
    $mail->Port = 465;
    $mail->SMTPAuth = true;
    $mail->SMTPSecure = PHPMailer::ENCRYPTION_SMTPS;
    $mail->SMTPDebug = 0; // 2 permet d'avoir le debuggage des
    $mail->Username = 'sandrineECF@laposte.net';
    $mail->Password = 'sandrineECF1234&';
    $mail->Helo = 'DEV-PC';
    $mail->AuthType = 'LOGIN PLAIN';
    $mail->SMTPOptions = array(
        'ssl' => array(
            'verify_peer' => false,
            'verify_peer_name' => false,
            'allow_self_signed' => true
        )
    );
}

```

1. Formulaire de Demande d'Information

Interactivité Client : Un formulaire est disponible pour les clients souhaitant obtenir des informations supplémentaires sur un véhicule spécifique.

Ce formulaire capte la référence du véhicule et les coordonnées du client pour permettre une communication efficace entre le garage et le client.

Formulaire de contact

Nom

Prénom

email

Téléphone (0001000000)

Votre adresse postale complète

Votre message

☐ En cliquant sur "Envoyer" j'accepte les conditions d'utilisation

Envoyer

Horaires d'ouverture

Lundi :

Mardi :

Mercredi :

Jeudi :

Vendredi :

Samedi :

Dimanche :

8h45 - 12h00 / 14h00 - 18h00

8h45 - 12h00 / 14h00 - 18h00

8h45 - 12h00 / 14h00 - 18h00

8h45 - 12h00 / 14h00 - 18h00

8h45 - 12h00 / 14h00 - 18h00

8h45 - 12h00

Faillir

Vous contactez :

Retrouvez nous :

© 2023 Company, Inc

Extrait du code formulaire :

```

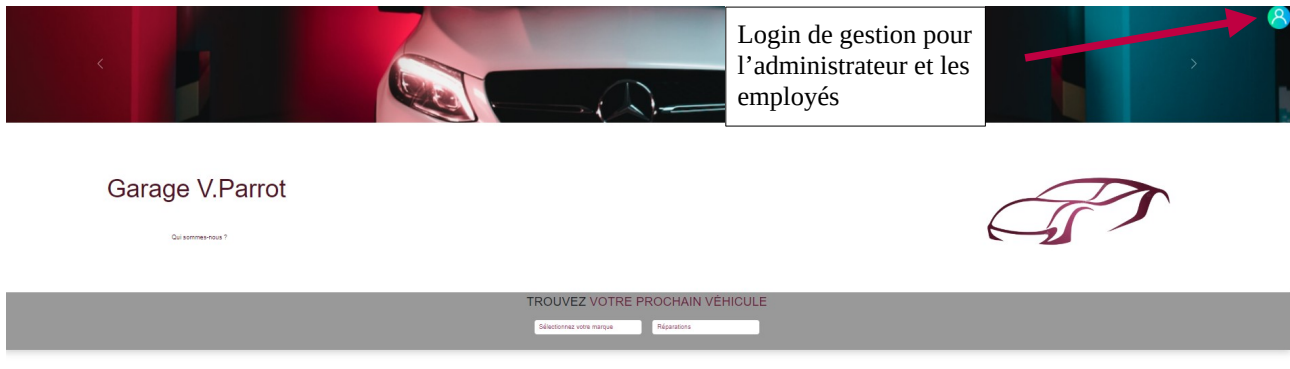
web > contact.php > body > script
16  /**
17  * Acquisition des datas du formulaire POST si mode soumission sinon les variables sont initialiées à null
18  *
19  * objectif affichage de la page BODY en bas
20  */
21  $nom      = POST::get('nom');
22  $prenom   = POST::get('prenom');
23  $email    = POST::get('email');
24  $phone    = POST::get('phone');
25  $adresse  = POST::get('adresse');
26  $message  = POST::get('message');
27  $valid_cgu = POST::get('valid_cgu');
28
29  /**
30  * Initilisation de variable de traitements
31  * exemple zone de focus / libelle de message d'erreur
32  */
33  $mesErr = '';
34  $zoneFocus = "";
35  /**
36  * Processus normalement lié à la validation suite à l'action de cliquer sur le bouton se connecter
37  */
38  $idxVehicule = null;
39
40  // acquisition en mode GET via les fiches véhicules pour la prise de contact via le <a href
41  if ($_SERVER['REQUEST_METHOD'] == 'GET'){
42      $idxVehicule = GET::get('code_vehicule');
43  }
44
45  // acquisition du formulaire
46  if ($_SERVER['REQUEST_METHOD'] == 'POST'){
47      /**
48      * Traitement des zones nom prenom email password adresse
49      */
50      $idxVehicule = POST::get('code_vehicule');
51
52      // les contrôles a faire à minima sur les zones
53      if ($nom == ''){
54          $mesErr = "La saisie du nom est obligatoire";
55          $zoneFocus = 'nom';
56      }

```

2. Gestion Administrateur et Employés

Comptes Sécurisés : Le site dispose d'un système de comptes avec mot de passe (chiffré en base de donnée) pour l'administrateur et les employés du garage.

Ceci leur permet de gérer les informations sur le site de manière sécurisée.



Fonctionnalités de Gestion : Les employés et l'administrateur ont la capacité de supprimer les annonces de voitures vendues et d'ajouter de nouveaux véhicules au catalogue ainsi que la publication ou non des véhicules sur le site web afin de les proposer en occasion (évidemment avec des statuts de réservation / vendu / annulation) pour permettre de ne plus montrer un véhicule si une demande d'information concernant un véhicule à donner suite à un rdv client pour action finale.

Liste des véhicules							
Marque	Description	Prix	Année Circulation	Km	Fichier Image	Action	Statut
ABARTH	FIAT	12990.00	2022	25000	FIAT_500_GREEN_680x430.jpg	Modif Supp.	Vendu ▾
ABARTH	ABARTH	13500.00	2020	32000	526_380.png	Modif Supp.	Réserver ▾
AUDI	BMX	40000.00	2020	10000	bmw blanche slider.jpg	Modif Supp.	Vendu ▾
BMW	AUDI	45500.00	2021	11000	Audi bleu slider.jpg	Modif Supp.	Arrivée ▾
FIAT	AUDI	53000.00	2022	11000	Audi cabriolet.avif	Modif Supp.	Publié ▾
FORD	Kia	10000.00	2019	32000	Kia jaune.avif	Modif Supp.	Publié ▾
MERCEDES	Cherokee	48000.00	2021	35000	Cherokee.png	Modif Supp.	Publié ▾
MINI	Mercedes	80000.00	2019	70000	Mercedes cabriolet.avif	Modif Supp.	Publié ▾
PEUGEOT	Range Rover	60000.00	2019	60000	Range-Rover marron.avif	Modif Supp.	Publié ▾
RENAULT	Renault Jaune	18000.00	2020	36000	Renault jaune.png	Modif Supp.	Publié ▾

Marque :

Description :

Prix :

Année de circulation :

Km :

Url Img : Aucun fichier choisi

3. Mise à Jour du Catalogue de Véhicules

Ajout et Suppression de Véhicules : Le site offre une interface facile à utiliser pour la mise à jour du catalogue de véhicules. Cela inclut l'ajout de nouvelles voitures et la suppression des annonces de véhicules vendus.

Et possibilité via jQuery de changer le statut du véhicule permettant ainsi de le publier sur les pages, de le mettre en statut de réservation vis à vis d'une demande par exemple et par voit de conséquence qu'il ne soit plus visible au sein du site, ensuite Vendu ou annuler (donc remis sur le site en publication).

Exemple javascript :

```
class ClassTemoignage {
  static publie(obj, idTemoignage) {
    $.ajax({
      url: "gestionTemoignage.php",
      data: "fct=publie&idTemoignage=" + idTemoignage,
      type: "post",
      success: function (myObj) {
        var objInput = obj.getElementsByTagName("INPUT");
        if (objInput) objInput[0].checked = myObj.is_publie ? true : false;
      },
    });
  }
  static interdit(obj, idTemoignage) {
    $.ajax({
      url: "gestionTemoignage.php",
      data: "fct=interdit&idTemoignage=" + idTemoignage,
      type: "post",
      success: function (myObj) {
        var objInput = obj.getElementsByTagName("INPUT");
        if (objInput) objInput[0].checked = myObj.is_interdit ? true : false;
      },
    });
  }
  static change(obj, idTemoignage) {
    $.ajax({
      url: "gestionTemoignage.php",
      data: "fct=change&idTemoignage=" + idTemoignage,
      type: "post",
      success: function (myObj) {
        document.getElementById("idCommentaire").innerHTML = myObj.commentaire;
      },
    });
  }
}
```


4. Gestion des Comptes Utilisateurs et Fonctionnalités Administratives

Comptes Administrateur et Employés

Comptes Personnels Sécurisés : Le site dispose d'un compte personnel pour l'administrateur V.Parrot et les employés, chacun protégé par un login et un mot de passe unique.

Gestion des Employés par l'Administrateur : L'administrateur a la capacité de gérer les comptes des employés, incluant la création, la modification et la suppression des comptes.

Fonctionnalités de Gestion des Véhicules

Panneau de Contrôle Administrateur : L'administrateur a accès à un tableau de bord permettant de gérer l'ensemble des fonctionnalités liées aux véhicules vendus, y compris la visualisation des ventes, la mise à jour des annonces et la gestion des stocks de véhicules.

Gestion des Annonces par les Employés : Les employés ont accès à leur propre interface pour gérer les nouvelles annonces de voitures. Cette fonctionnalité leur permet d'ajouter, de modifier ou de supprimer des informations sur les véhicules en vente, conformément au cahier des charges. De plus, ils agissent en tant que modérateurs sur les avis clients, pouvant décider d'afficher ou non un témoignage clients.

Sécurité et Accès

Sécurisation des Données : Des mesures de sécurité sont mises en place pour assurer la protection des données des comptes utilisateurs et des informations sensibles relatives aux véhicules :
Exemple ;chiffrement du mot de passe utilisateur.

```

/**
 * sécurisation du mot de passe SHA1 avec clé dynamique rajouté pour éviter le Brut force
 */
private static function pw_encode($password){
    $seed = "";
    for ($i = 1; $i <= 10; $i++)
        $seed .= substr('0123456789abcdef', rand(0,15), 1);
    return sha1($seed.$password.$seed).$seed;
}

private static function pw_check($password, $stored_value){
    if (strlen($stored_value) != 50)
        return FALSE;
    $stored_seed = substr($stored_value,40,10);
    return (sha1($stored_seed.$password.$stored_seed).$stored_seed == $stored_value);
}

```

Au sein de la classe classUtilisateur.php

Restrictions d'Accès : Des niveaux d'accès différenciés sont établis pour les administrateurs et les employés, assurant ainsi que chaque utilisateur ait accès uniquement aux fonctionnalités nécessaires à son rôle exemple de méthode dans la classUtilisateur.php

```

/**
 * Renvoi true false pour savoir si l'utilisateur est admin
 * @return boolean
 */
public function isAdmin(){
    return $this->user['type_utilisateur'] == 'A';
}

```

Objectif du Système de Gestion

Cette structure de gestion a pour but de faciliter la gestion interne du garage, en offrant une plateforme efficace et sécurisée pour la manipulation des données des véhicules et la gestion des employés.

Elle vise à optimiser les opérations quotidiennes et à améliorer l'efficacité administrative du garage.

5. Configuration de l'environnement de travail

Liste des logiciels nécessaires

1. Navigateurs Web

Compatibilité Multi-Navigateurs : Site compatible avec les principaux navigateurs web ; Google Chrome, Mozilla Firefox, Safari, Microsoft Edge.

2.Éditeur de Code

Visual Studio Code (VSCode) : L'éditeur choisi pour le développement du projet et ses nombreuses extensions, son intégration Git, et son support robuste pour le HTML, CSS, JavaScript, et PHP.

Serveur Local

3.WAMP : Utilisé pour simuler un environnement de serveur local sur les machines Windows. WAMP est crucial pour le développement et le test des fonctionnalités PHP et MySQL du site avant son déploiement sur le serveur de production.

Versions des applications utilisé :

MYSQL 8.0

PHP 8.026

Apache 2.4.54

Langage et frameworks

Environnement Front-end

1.HTML : Version HTML5 recommandée pour assurer la compatibilité avec les normes web modernes.

2.CSS : Utilisation de CSS3 pour les styles et les mises en page, compatible avec les principaux navigateurs.

3.JavaScript : Version ECMAScript 6 (ES6) ou ultérieure, pour une programmation plus efficace et moderne.

3.jQuery : Version spécifique utilisée (par exemple, jQuery 3.5.1), pour garantir la compatibilité des plugins et des fonctionnalités.

4.Bootstrap : Utilisation de Bootstrap (5.3.1) pour le design réactif et les composants d'interface utilisateur.

```

13 </div>
14
15 <div class="paper_plane">
16   <a href="contact.php">Nous contacter</a>
17   
18   <span>Retrouvez nous</span>
19
20   
24 </div>
25
26
27 <div class="col-md-4 d-flex align-items-center">
28   <a
29     href="/"
30     class="my-6 mb-3 me-2 mb-md-0 text-body-secondary text-decoration-none lh-1"
31   >
32     <svg class="bi" width="30" height="24">
33       <use xlink:href="#bootstrap"></use>
34     </svg>
35   </a>
36   <span class="my-4 mb-3 mb-md-0 text-body-secondary">© 2023 Company, Inc</span>
37 </div>
38
39 </div>
40
41 <script src="web/app.js"></script>
42
43 </footer>
44

```

Environnement Back-end

PHP : Version 8

MySQL : Version 8

En règle général je choisie toujours la dernière version existante.

Installation et configuration

Serveur Web et PHP : Instructions pour installer et configurer WAMP, selon mes cours et d'autres sources d'informations sur le net.

MySQL avec Workbench : Guide pour installer MySQL et MySQL Workbench, création de la base de données. selon mes cours et d'autres sources d'informations sur le net.

Configuration du projet :

Dépôt Git des fichiers du projet par l'extension git dans VSCode avec un premier commit.

Configuration de l'environnement (ex : fichiers .env pour PHP):

Environnement de Production

Hébergement Web et Serveur de Base de Données :

Choix de l'Hébergement Web (pas encore déployé)

Mes critères de sélection :

Capacité de stockage et bande passante adaptée à la taille et au trafic prévu du site.

Compatibilité avec les technologies utilisées (PHP, MySQL).

Support technique fiable et temps de disponibilité élevé.

Options de sécurité, comme les certificats SSL (notamment pour mettre en place la coche HTTPS), pour protéger les données des utilisateurs (chiffrement des données).

Conseils pour choisir un hébergement adapté.

Arborescence des fichiers

- bibappli : Comprend des bibliothèques spécifiques comme 'lib_metier.php' permettant des fonctionnalités métiers.

- class : Contient des classes pour la gestion des clients, des utilisateurs et des véhicules, indiquant que l'administrateur peut gérer des utilisateurs, des demandes clients et des informations sur les véhicules.

- lib et lib_ext : Ces dossiers contiennent des bibliothèques générales et externes, respectivement. On note la présence de PHPMailer, permettant des fonctionnalités d'envoi de mails.

- lib_page : Possède des fichiers qui sont utilisés pour la structure de la page, comme le header et le footer.

- web : Le dossier principal pour les ressources web. Il a des sous-dossiers pour les assets, CSS, images et JavaScript. Le fichier 'signin.php' permettant une fonction d'authentification.

Description des modules : Expliquant brièvement chaque module, sa fonction et son interaction avec les autres et celui en dehors du web (à faire)**

- index.php : page de garde avec sélecteur de marque et réparation (dev à faire), avec aussi présentation des véhicules de moins de 20 000 euro et faible Km qui sont indiqués comme publiable sur la vitrine.

- voitures-occasions.php : qui affiche une liste de véhicules concernant une marque et étant publiable sur la vitrine.
- fiche_voiture.php : qui affiche les détails d'un véhicule avec au sein de cette page un lien permettant d'accéder à la page contact.php.

Lors de l'affichage d'un véhicule, en interne, l'usage de la fonction vignette_auto.php est déclencher avec au sein de cette page un lien permettant d'accéder à la page contact.php.

- contact.php qui permet de saisir via un formulaire une demande qui sera enregistré en base associé à un véhicule et envoi un mail au demandeur pour lui indiqué que c'est ok.
- signin.php : formulaire permettant de s'authentifier
- utilisateur.php : gestion des utilisateurs (écran uniquement accessible par un utilisateur admin, dans le cas contraire, si un appel est fait sur cette page, la page index.php est relancé).
- vehicule.php : gestion des véhicules (écran uniquement accessible dès lors qu'un utilisateur est connecté puisque disponible pour tout type d'utilisateurs connecté).
- gestionDemandeClient.php : gestion des demandes clients saisie via la fiche contact.php (écran uniquement accessible dès lors qu'un utilisateur est connecté puisque disponible pour tout type d'utilisateurs connecté).
- gestionTemoignage.php : gestion des témoignages clients, permet la publication des témoignages ou le rejet de celui ci (écran uniquement accessible dès lors qu'un utilisateur est connecté puisque disponible pour tout type d'utilisateurs connecté).

Configuration Locale avec WAMP64

Utilisation de WAMP64 : En développement local, le projet utilise WAMP64 qui inclut Apache, MySQL, et PHP.

Alias Apache : Un alias a été créé dans Apache de WAMP64 pour faciliter l'accès local au projet.

Rôle de WAMP64 : WAMP64 sert à simuler un environnement de serveur et à tester le site en conditions réelles sur votre machine locale.

Gestion de la Base de Données MySQL

Création de Tables MySQL : Au lieu d'utiliser PHPMysqlAdmin, les tables MySQL ont été créées via cet outil.

Cette approche nécessite une compréhension approfondie de la structure relationnelle des bases de données dans l'approche MCD / MPD.

Instructions pour la Création des Tables : Fournissez un script SQL ou des instructions détaillées sur la manière de créer les tables nécessaires pour le site.

Guide de Développement

HTML/CSS : Normes de codage, frameworks utilisés, et organisation des fichiers.

JavaScript et jQuery : Conventions de codage, utilisation de bibliothèques externes.

PHP : Architecture (MVC si utilisé), gestion des dépendances, sécurité.

Architecture

Modèle de Conception : Le site n'utilise pas explicitement l'architecture MVC (Modèle-Vue-Contrôleur), mais suit un modèle structuré pour séparer la logique métier (traitement des données), la présentation (HTML/CSS) et les interactions avec la base de données.

Fichiers PHP : Les fichiers PHP sont organisés de manière à séparer clairement les requêtes de base de données, la logique de traitement et la présentation. Par exemple, les scripts de connexion à la base de données sont isolés dans des fichiers distincts pour une meilleure maintenance et sécurité.

Gestion des Dépendances

Bibliothèques et Frameworks : Le projet n'emploie pas de gestionnaire de dépendances complexe comme Composer, mais utilise des bibliothèques tierces (telles que jQuery et Bootstrap) intégrées directement dans les fichiers.

Inclusion de Fichiers : Les fichiers PHP utilisent des instructions `include` ou `require` ou `require_once` pour gérer les dépendances internes, permettant de réutiliser le code et de maintenir une structure organisée.

Sécurité

Validation des Données : Toutes les données entrantes (formulaires, requêtes GET/POST) sont validées et échappées pour éviter les injections SQL et les attaques XSS. Utilisation de méthode `quote` de pdo.

Authentification : Le système de gestion des comptes administrateur et employés implique une authentification sécurisée. Les mots de passe sont stockés sous forme hashée dans la base de données MySQL.

Gestion des Sessions : Les sessions PHP sont utilisées pour gérer les connexions utilisateur. Des mesures sont prises pour sécuriser les sessions contre les détournements et les attaques par fixation de session.

Contrôle d'Accès : Des vérifications de droits d'accès sont effectuées pour s'assurer que seuls les utilisateurs authentifiés peuvent accéder aux fonctionnalités administratives.

Sécurité des Fichiers et Serveur : Des précautions sont prises pour protéger les fichiers sensibles (par exemple, fichiers de configuration PHP) contre l'accès direct. Des configurations de serveur appropriées sont utilisées pour renforcer la sécurité globale.

MySQL : Modèle Conceptuel de Données, relations entre les tables, scripts d'initialisation.

Guide de Déploiement (pas encore déployé)

Procédures de déploiement : Étapes pour déployer l'application sur un serveur de production.

Gestion des versions : Utilisation de Git pour la gestion des versions et des branches.

Tests et Maintenance

Tests : Outils et méthodes de tests recommandés (unitaires, fonctionnels).

Débogage : Outils et techniques de débogage.

Manuel du projet

Cette documentation fournit une vue d'ensemble complète du projet, en couvrant les aspects techniques, de conception et d'utilisation. Elle est conçue pour guider le correcteur à travers le projet, depuis la compréhension des choix technologiques jusqu'à l'installation et le test des fonctionnalités.

6. Mise en œuvre

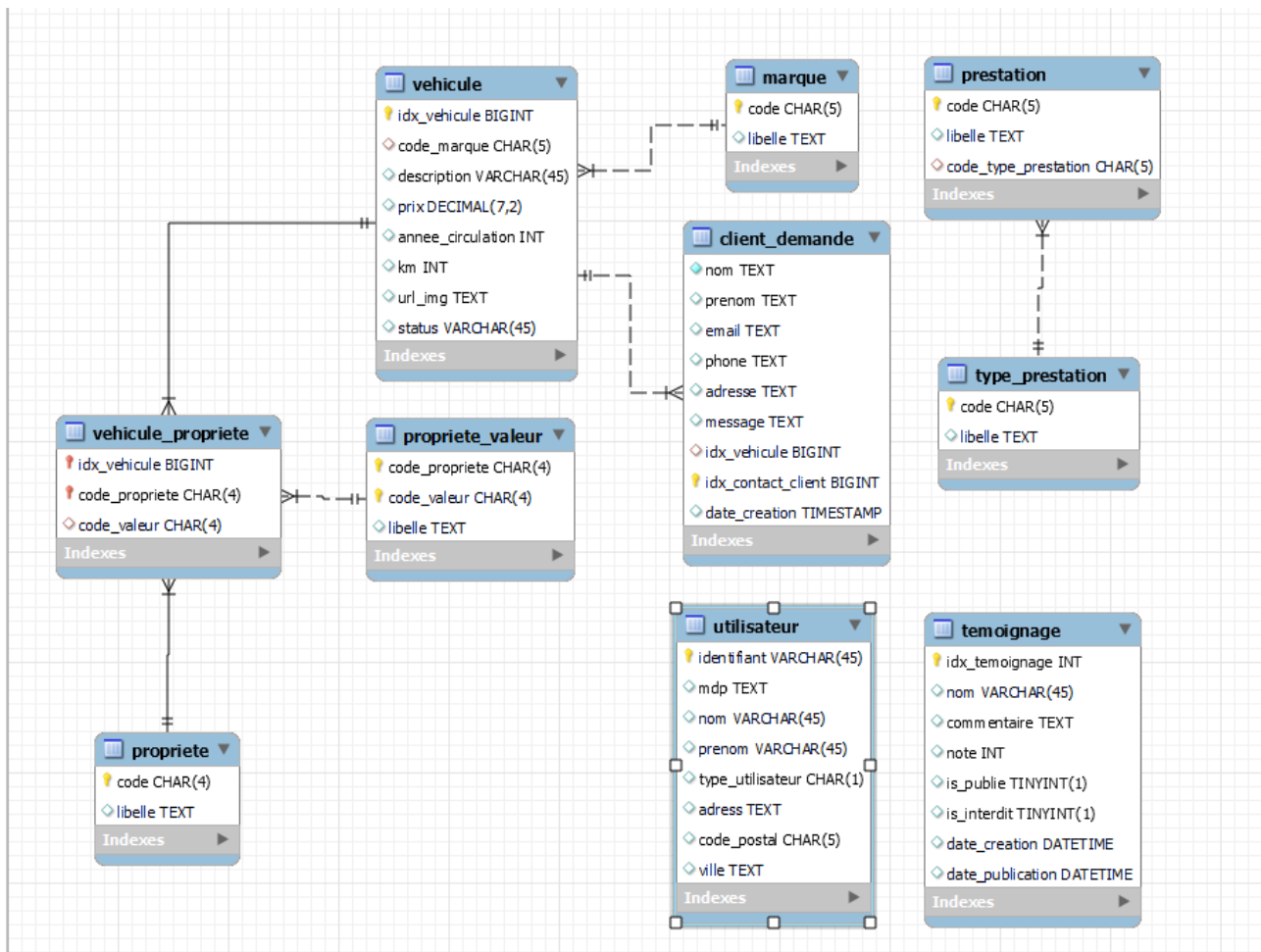
1. Choix des Technologies
2. Diagrammes et Conception
3. Charte Graphique
4. Manuel d'Installation et de Test

1. Choix des Technologies

- Front-End : HTML5, CSS3, JavaScript, jQuery, Bootstrap
- Back-End : PHP, MySQL (avec WAMP64)
- Environnement de Développement : Visual Studio Code, WAMP64
- Sécurité : Authentification des utilisateurs, validation des données, gestion sécurisée des sessions

2. Diagrammes et Conception

- Wireframes : Présentation des wireframes de l'interface utilisateur, montrant la disposition des éléments sur les principales pages.
- Diagramme de Classe : Structure de la base de données et relations entre les tables.



Partant de cela, au niveau de Php, l'on retrouve les classes `classUtilisateur`, `classTemoignage`, `classVehicules`, `classClientDemande`, qui contiennent les méthodes par exemple d'ajout, de modification, suppression ... (CRUD)

3. Charte Graphique

- Couleurs et Typographie : Détail des couleurs, polices et styles utilisés pour assurer la cohérence visuelle du site- Éléments Graphiques** : Description des logos, images, icônes et autres éléments graphiques.

** Vous trouverez ma charte graphique à part, car le format cassé l'ensemble du document,

4. Manuel d'Installation et de Test

Installation en Local

1. Configuration de WAMP64

WAMP (Windows, Apache, MySQL, et PHP) est un logiciel qui permet de créer un environnement de serveur local sur un ordinateur Windows. Voici comment le configurer :

a. Téléchargement et Installation

Rendez-vous sur le site officiel de WAMP (wampserver.com).
Téléchargez la version qui correspond à votre système (32 bits ou 64 bits).
Exécuter le fichier d'installation et suivre les instructions.

b. Démarrage de WAMP

Lancez WAMP depuis le menu Démarrer.
Une fois démarré, une icône de WAMP apparaîtra dans la barre des tâches (elle devrait être verte indiquant que tous les services sont actifs).

c. Test

Ouvrir un navigateur web et tapez localhost dans la barre d'adresse. Si la page d'accueil de WAMP s'affiche, l'installation est réussie.

- Mise en place de la base de données MySQL.

Gestion de MySQL sans phpMyAdmin

1. Accès à MySQL via la Ligne de Commande

a. Ouvrir le Terminal MySQL

Allez dans le dossier où WAMP est installé (par défaut, il s'agit de C:\wamp64).

Accédez au sous-dossier bin\mysql\mysqlx.x.x\bin (remplacez mysqlx.x.x par la version de MySQL installée avec votre WAMP).

Maintenez la touche Shift enfoncée et faites un clic droit dans le dossier, puis choisissez Ouvrir la fenêtre de commande ici ou Ouvrir la fenêtre PowerShell ici.

Tapez mysql -u root -p (si vous n'avez pas défini de mot de passe pour l'utilisateur root, laissez le champ du mot de passe vide).

2. Exécution de Scripts SQL

a. Création de la Base de Données

Tapez CREATE DATABASE nom_de_votre_db; pour créer une nouvelle base de données.

b. Sélection de la Base de Données

Tapez USE nom_de_votre_db; pour sélectionner votre base de données.

c. Création de Tables et Manipulation des Données

Utilisez des commandes SQL standard pour créer des tables, insérer des données, etc. Par exemple, CREATE TABLE, INSERT INTO, SELECT, UPDATE, etc.

3. Utilisation de Scripts SQL depuis un Fichier

a. Création d'un Fichier SQL

Créez un fichier .sql contenant vos commandes SQL.

Dans n'importe quel éditeur de texte.

b. Exécution du Script

Dans le terminal MySQL, utilisez la commande source chemin_vers_votre_fichier.sql; pour exécuter les commandes SQL contenues dans votre fichier.

4. Intégration avec VSCode

a. Exécution des Scripts SQL depuis VSCode

Vous pouvez ouvrir le terminal directement dans VSCode (Terminal > Nouveau terminal).
Naviguez jusqu'au dossier MySQL comme indiqué précédemment.

Exécutez vos scripts SQL en utilisant la commande source ou en tapant manuellement les commandes.

Avantages de cette Méthode

Contrôle complet : Vous avez un contrôle total sur les opérations de base de données.

Pratique des commandes SQL : C'est un excellent moyen de s'habituer à écrire et à comprendre les commandes SQL.

Automatisation : Facilite l'automatisation de certaines tâches via des scripts.

Cette approche est idéale si vous souhaitez avoir un contrôle complet sur votre base de données et améliorer vos compétences en SQL.

- Configuration du projet dans VSCode.
- Guide de Test :
- Processus pour tester les fonctionnalités clés (ex: visualisation des véhicules, utilisation des formulaires, gestion par l'administrateur).
- Scénarios de test pour vérifier le bon fonctionnement du site.