

#LIVRABLES

##Fichier de création et d'alimentation de la base de donnée :

****Documentation technique****

A. INTRODUCTION

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 :

1. Catalogue de Voitures d'Occasion

Base de Données des Véhicules : Le site intègre une base de données MySQL contenant une liste de marques de voitures avec des informations détaillées sur chaque véhicule

(modèle, année, kilométrage, prix, etc.).

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

2. 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.

3. Gestion Administrateur et Employés

Comptes Sécurisés : Le site dispose d'un système de comptes avec mot de passe 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.

4. 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.

5. Gestion des Comptes Utilisateurs et Fonctionnalités Administratives

a) 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.

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.

****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.

B. 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.

LANGAGES 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.x), pour garantir la compatibilité des plugins et des fonctionnalités.
- 4.Bootstrap : Utilisation de Bootstrap (préciser la version, par exemple Bootstrap 4.x ou 5.x) pour le design réactif et les composants d'interface utilisateur.

****Environnement Backend****

PHP : Version 7.x recommandée pour profiter des améliorations de performance et de sécurité.

MySQL : Version 5.x, compatible avec WAMP et les outils de développement utilisés.

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, pour protéger les données des utilisateurs.

Conseils pour choisir un hébergement adapté.

3. Structure du Projet

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 'authentication.php' permettant une fonction d'authentification.

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

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 PHPMyAdmin, les tables MySQL ont été créées manuellement. Cette approche nécessite une compréhension approfondie de la structuration des bases

de données et des commandes SQL.

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.

4. 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 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 nettoyées pour éviter les injections SQL et les attaques XSS. Utilisation de fonctions comme htmlspecialchars et mysqli_real_escape_string.

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 de données, relations entre les tables, scripts d'initialisation.

5. 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.

6. Tests et Maintenance

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

Débogage : Outils et techniques de débogage.

UNE IDÉE SUPPLÉMENTAIRE pour amélioration

7. Annexes

Ressources supplémentaires : Liens vers des ressources externes, documentation.

FAQ : Réponses aux questions fréquemment posées.

8. Contact et Support

Contacts : Informations pour contacter l'équipe du projet.

Support : Où et comment obtenir de l'aide.

****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.

Sommaire

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.
- **Diagramme de Cas d'Utilisation** : Illustration des interactions des utilisateurs avec le système.
- **Diagramme de Séquence** : Flux des opérations pour les processus clés (ex: ajout de véhicule, demande d'information).

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.

4. Manuel d'Installation et de Test

- **Installation en Local** :
 - Configuration de WAMP64.

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 votre 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.