

SOMMAIRE :

- Début
- Maquette
- Écriture fonctionnalités
- Diagrammes
- Création base de donnée
- Création projet
- Connexion
- Vérifier utilisateur connecté
- Vérifier rôle Admin
- Inscription
- Header
- Footer
- Création du bandeau
- Afficher le bandeau
- Écrire un commentaire
- Afficher les commentaires
- Page d'accueil
- Page de réparations
- Page de vente
- Page de contact
- Gestion
- Vendre véhicule
- Supprimer annonce
- Modifier annonce
- Modifier les horaires
- Commentaires en attente
- Commentaires
- Atelier formulaire
- Vente formulaire
- Contact
- Réparation
- Inscrire un employé
- Modifier utilisateur
- Supprimer utilisateur
- Héberger le site
- Export base de donnée

DÉBUT

Pour commencer ce projet la, j'ai tout d'abord fait sur un morceau de papier un wireframe pour avoir une idée global de a quoi pourrais ressembler le site web, j'ai fais quelques modifications dessus pour avoir un visuel plus attractif et agréable pour les yeux des potentielles visiteurs.

MAQUETTE

J'ai mis tout au propre pour faire une maquette sur Figma, pour que je puisse avoir un vrai point de vue et idée global de mon site web.

ÉCRITURE FONCTIONNALITÉS

Par la suite, j'ai noté sur un papier toutes les fonctionnalités dont j'allais alors besoin pour ce projet, que ce soit un simple header ou bien des composants plus complexe comme la création de compte ou alors le filtrage de voitures sans faire de rafraîchissement de page web.

DIAGRAMMES

Une fois mon cahier des charges écrit, j'ai commencé a faire les diagrammes pour avoir une vue global sur ma base de donnée et ses différentes tables nécessaires au bon fonctionnement de l'application web et de toutes les fonctionnalités dont j'ai besoin. J'ai donc fais un diagramme de séquence, diagramme de cas d'utilisation et méthode merise.

CRÉATION BASE DE DONNÉE

Une fois tous mes diagrammes créé, j'ai créé ma base de donnée. J'ai ensuite fais des requêtes SQL (toutes disponibles dans le fichier « requeteSql.sql ») pour créer mes différents tables au sein de ma base de donnée. Les 10 tables sont « bandeau_reparation, commentaires, commentaires_attente, formulaire_atelier, formulaire_contact, formulaire_vente, horaires, reparation, users, voitures ».

CRÉATION PROJET

Une fois fait, j'ai initié le projet sur l'IDE VisualStudio Code car j'aime beaucoup cet IDE. J'ai commencé par créer mon composant qui va me servir à me connecter à ma base de donnée en utilisant PDO.

CONNEXION

Pour le système de connexion, j'ai un formulaire avec les champs « e-mail, mot de passe ». Je vais vérifier que tous les champs soit bien remplies et si c'est le cas, j'importe ma base de donnée et je vais faire une requête SQL de mes utilisateurs par rapport à l'e-mail renseigné dans le champs e-mail. Si l'e-mail est bon, on va venir récupérer le mot de passe associé a l'adresse e-mail et le vérifié, si tout correspond, l'utilisateur est alors connecté. On va alors créer sa session et récupérer des informations, telle que son « rôle ». Dès qu'il est connecté, l'utilisateur va automatiquement être redirigé sur la page de gestion.

VÉRIFIER UTILISATEUR CONNECTE

Par la suite, j'ai créé le script permettant de vérifier si un utilisateur est bien connecté. Pour ce faire, je viens récupérer les données de la super global \$_SESSION qui est créé lors qu'un utilisateur se connecte, si l'utilisateur est bien connecté, il peut accéder aux pages dont il a accès. S'il n'est pas connecté, l'utilisateur du site web sera bloqué sur certaines pages.

VÉRIFIER RÔLE ADMIN

Après, j'ai créé le système pour vérifier si le rôle de l'utilisateur correspond à « Admin ». Pour ce faire, je récupère sur la super global \$_SESSION son rôle, si le rôle correspond à « Admin », il pourra alors accéder aux pages dont il a l'autorisation. Dans le cas contraire, l'utilisateur sera bloqué sur certaines pages.

INSCRIPTION

Pour le système d'inscription d'utilisateur, j'ai créé un formulaire avec 3 champs « nom, e-mail, mot de passe » qui sont tous required. Une fois les inputs remplies, les valeur sont stockés dans des variables et sécurisé. Pour le mot de passe, j'utilise une fonction permettant de hasher le mot de passe en base de donnée pour protéger le mot de passe des utilisateurs inscrit au sein de l'application web. J'envoie toutes les valeurs dans ma base de donnée sur la table users. Pour cette page, j'ai importé un script permettant de vérifier le rôle de l'utilisateur soit bien « Admin ». A savoir que tous les nouveaux utilisateurs inscrit auront automatiquement le rôle « Employé ».

HEADER

J'ai ensuite créé mon header avec le nom du garage et les différents liens de navigations sur les pages du site web. Le fichier header contient tout le head sur HTML, toutes les balises méta et les importations des mes fichiers CSS.

FOOTER

Pour le footer, j'ai choisi d'y afficher les horaires du site web. Pour ce faire, j'ai importé ma base de donnée dans mon composant. J'ai fait une requête SQL pour sélectionner toute ma table contenant tous les horaires et j'ai fait une boucle foreach pour tous les afficher dans le footer. Pour compléter le footer, j'y ai intégré des liens factices vers les réseaux sociaux du garage, ainsi qu'un numéro de téléphone pour pouvoir leur téléphoner.

CRÉATION DU BANDEAU

J'ai après créé le bandeau de réparation qui va servir à montrer aux visiteurs de l'application web, les différents types de réparations que le garage est habilité à effectuer sur les véhicules clients. Pour ce faire, j'ai du créer un formulaire pour y injecter des valeurs et photos en base de donnée. Un bloc de ce bandeau contient un titre et une photo. Pour envoyer une photo en base de donnée, j'ai utilisé un input type file, l'image est alors stocké au format LONGBLOB.

AFFICHER LE BANDEAU

Pour afficher le bandeau de réparation, j'ai fait une requête SQL pour y récupérer toutes mes lignes dans la table qui stock ces informations. J'ai fait une boucle pour pouvoir tout afficher sans avoir besoin de faire beaucoup de ligne de code. Pour pouvoir afficher mes images, j'ai du utiliser « base64_encode » car au lieu d'avoir mon image, j'avais un texte à la place. J'ai mis toutes les lignes dans un container pour pouvoir donner du style à mon bandeau.

ÉCRIRE UN COMMENTAIRE

Pour l'ajout de commentaire, j'ai fait un formulaire avec 3 champs « nom, commentaire, note ». Je récupère dans des variables les valeurs des inputs renseigné par l'utilisateur pour sécurisé les donnée saisie. Une fois fait, je fais une requête SQL pour envoyer dans ma table contenant les commentaires en attente, les données de mes différentes variables.

AFFICHER LES COMMENTAIRES

Pour afficher les commentaires, j'ai créé un container qui va servir à y mettre tous les commentaires. Je fais une requête SQL pour récupérer tous les commentaires validé et je fais une boucle foreach pour tous les afficher dans mon container.

PAGE D'ACCUEIL

Pour la premier page du site, j'ai importé le header et j'ai décidé d'y afficher le logo du garage en dessous du header, car le logo est l'image du garage. J'ai aussi eu envie de faire une courte description qui parle de Vincent et du garage. Cette description sert à parler de Vincent, de ses expériences et de ses compétences en tant que garagiste. J'ai ensuite importé mon bandeau de réparation pour que l'utilisateur puisse avoir plusieurs informations sur le garage sans devoir forcément naviguer plusieurs minutes sur le site web avant de voir les types de réparations possible. J'ai ensuite importé le script permettant de laisser un commentaire sur le site web, et en dessous, le script permettant d'afficher tous les avis du site web. Et pour le pied de page, j'ai fini par importé le footer.

PAGE DE RÉPARATION

Pour la page réparation, j'ai importé mon header puis mon bandeau de réparation. J'ai également créé un récapitulatif de mon bandeau, j'ai fais une description de chaque type de réparation avec le titre associé. Les descriptions viennent directement de ma base de donnée pour que l'admin puisse le modifier s'il décide de changer un type de réparation pour un autre. J'ai ensuite mis en dessous un formulaire de contact spécialement pour l'atelier. Ce formulaire comporte les champs « nom, prénom, e-mail, téléphone, raison, message ». Pour le champs « raison », j'ai mis un input de type select et j'ai mis toutes les lignes de ma table bandeau_reparation pour que l'utilisateur choisisse obligatoirement une réparation disponible par le garage. Une fois le formulaire remplie, je viens vérifier que le formulaire et les champs soient correctement remplies et non vide. Si c'est le cas, je stock les valeurs des inputs dans des variables et je viens préparer ma requête SQL pour insérer en base de donnée les données du formulaire. Et pour finir, j'importe le footer pour le pied de page.

PAGE DE VENTE

Pour la page de vente, j'ai commencé par créer mon container que va contenir mes voitures venant de ma table pour les véhicules. Ensuite, j'ai créé ma requête SQL pour récupérer toutes mes voitures. J'ai mis en place un système de filtrage pour les valeurs « prix, année, kilométrage » dans des inputs de type range. Suivant les valeurs sélectionnées par les utilisateurs, des voitures qui rentrent dans les critères s'afficheront. Pour ce faire, j'ai utilisé du Java Script mais plus particulièrement des requêtes AJAX. Dans un premier temps, je récupère la valeur de mon input et j'affiche donc la valeur dans un span pour montrer à l'utilisateur qu'elle valeur est renseigné actuellement par l'input. J'importe jQuery et je prépare la requête Ajax, mes voitures sont alors récupéré du fichier « fetch_cars.php » pour les afficher dans une div. Une fois les voitures affichées, j'ai mis en place un bouton pour pouvoir arriver sur une nouvelle page généré automatiquement pour avoir la carte de la voiture, mais aussi un formulaire de contact pour la voiture. Le formulaire possède 3 champs « voiture, nom, téléphone », le champs « voiture » est renseigné automatiquement par le nom du véhicule en question. Une fois le formulaire remplie, je viens à le vérifier s'il est bien remplie et si c'est le cas, je stock les valeurs dans des variables pour tout envoyer en base de donnée dans la table qui correspond au formulaire de contact pour les véhicules.

PAGE CONTACT

Pour la page contact, j'ai d'abord importé mon header et mis en place un formulaire mais qui cette fois est pour contacter le garage directement. Ça peut être par exemple pour signaler un problème sur le site web, ou une demande de stage. Ce formulaire contient plusieurs champs « nom, prénom, e-mail, téléphone, raison, message », tous ces champs sont obligatoirement. Comme pour tous les formulaires du site web, le formulaire est vérifié entièrement et je viens stocké dans des variables les valeurs renseignées dans les inputs. Une fois fais, je prépare la requête SQL et j'envoie en base de donnée dans la table qui correspond, les données saisie par l'utilisateur. Et pour finir, je viens importer le footer pour le pied de la page.

GESTION

Pour cette page, j'ai tout d'abord créé une « barre de navigation » pour naviguer entre toutes les différentes pages du panel de contrôle. Il y a un total de 14 pages différentes « connexion, vendre véhicule, supprimer annonce, modifier annonce, modifier les horaires, commentaires en attente, commentaires, atelier formulaire, vente formulaire, contact, réparation, inscrire un employé, modifier utilisateur, supprimer utilisateur ». J'ai également créé un système pour afficher le nombre d'utilisateur, le nombre de formulaire pour chaque cas, et le nombre de commentaire, pour pouvoir avoir un point de vue global sans avoir forcément besoin d'ouvrir toutes les pages pour savoir s'il y a de nouvelles demandes de contact ou de nouveaux commentaires en attente. Pour ce faire, j'ai fais une requête SQL pour chaque donnée dont je voulais savoir le nombre, et j'ai précisé que je voulais récupérer le nombre de lignes dans la table en question. C'est page est accessible uniquement si un utilisateur est inscrit sur le site web. Cette page est accessible aux employés et à l'admin seulement.

VENDRE VÉHICULE

Dans cette page, j'ai simplement créé un formulaire pour pouvoir vendre des voitures, les champs sont « Nom de la voiture, kilométrage, année, transmission, cylindré, chevaux, prix, photo ». Les photos des voitures en base de donnée sont stocké au format LONGBLOB. Tous ces champs sont obligatoire. Une fois le formulaire remplie, je viens vérifier que tout soit bien remplie et que l'image fournit soit au bon format. Si c'est le cas, je stock dans des variables tous les champs qui ont été saisie et je les bind à la requête SQL. C'est page est accessible uniquement si un utilisateur est inscrit sur le site web. Cette page est accessible aux employés et à l'admin seulement.

SUPPRIMER ANNONCE

Pour cette page, le but est de gérer les annonces déjà mises en ligne. Dans un premier temps, je fais une requête SQL pour récupérer toutes les voitures en base de donnée pour les afficher dans un formulaire avec un bouton « delete » pour pouvoir en supprimer. J'ai mis en place le code de cette façon, je récupère la valeur du bouton qui a été cliqué, le bouton à comme valeur l'ID de la voiture. Après, j'utilise cette variable pour la lier à la nouvelle requête SQL pour supprimer seulement la voiture qui a cet ID là, et non toutes les voitures disponibles sur la page. Cette page est accessible uniquement si un utilisateur est inscrit sur le site web. Cette page est accessible aux employés et à l'admin seulement.

MODIFIER ANNONCE

Pour cette page, c'est la modification de voitures déjà mise en vente. Pour faire ça, j'ai récupéré de ma base de donnée, toutes les voitures en vente. J'ai fait une boucle foreach et affiché dans un formulaire, toutes les informations de mes voitures. Les inputs sont déjà remplis avec les valeurs déjà enregistrées des voitures. Pour changer celle qui a été sélectionnée, je récupère l'id du formulaire qui a été validé et le mets dans la requête SQL avec les informations saisies pour changer directement en base de donnée. Cette page est accessible uniquement si un utilisateur est inscrit sur le site web. Cette page est accessible aux employés et à l'admin seulement.

MODIFIER LES HORAIRES

Pour cette page, j'ai d'abord créé un composant pour changer les horaires de lundi. Pour ce faire, je récupère dans ma table horaires, la ligne qui correspond au lundi. Je crée un formulaire avec deux inputs « matin, apresmidi ». Ces inputs ont pour valeur la valeur déjà renseignée en base de donnée. Dès que l'admin va modifier un champ, je stocke cette valeur dans une valeur que j'intègre à la requête SQL pour changer les horaires du lundi. Pour les autres jours, c'est le même fonctionnement, j'ai au total 7 composants pour les 7 jours de la semaine. J'ai préféré faire ça en DUR car dans tous les cas, les jours de la semaine resteront à 7. J'importe les 7 composants sur la page pour avoir mes 7 formulaires sur une seule et même page. Cette page est accessible à l'admin seulement.

COMMENTAIRES EN ATTENTE

Pour cette page, on y reçoit tous les commentaires qui viennent d'être écrits par les utilisateurs. Pour les afficher, je les récupère de ma base de donnée et fais une boucle pour qu'ils soient tous disponibles sur la page. Il y a deux boutons disponibles, un pour valider le commentaire, et un autre pour le supprimer. Si le commentaire est validé, il est alors supprimé de la table mais est envoyé sur la table des commentaires validés. S'il est supprimé, il est totalement supprimé de la base de donnée. Cette page est accessible aux employés et à l'admin seulement.

COMMENTAIRES

Sur cette page, c'est très simple. Je récupère tous les commentaires qui ont été validé. Cette fonctionnalité sert à supprimer des commentaires qui ont déjà été validé. Par exemple si quelqu'un a laissé un commentaire mais qu'au final il ne veut pas apparaître sur le site, un employé peut directement effacer son commentaire. Cette page est accessible aux employés et à l'admin seulement.

ATELIER FORMULAIRE

Pour cette page, je viens récupérer tous mes formulaire de contact pour l'atelier. Je les récupère avec une requête SQL qui va me servir à sélectionner tous les formulaire présent dans la table en question. Je les affiche donc dans un container pour avoir de la lisibilité sur la page. La fonctionnalité de cette page est de supprimer le formulaire de contact. Par exemple si la demande a déjà pu être traité, l'employé ou l'admin aura juste à la supprimer pour laisser place à des demandes en attente de traitement. Cette page est accessible aux employés et à l'admin seulement.

VENTE FORMULAIRE

Pour cette page, je viens récupérer tous mes formulaire de contact pour les voitures. Je les récupère avec une requête SQL qui va me servir à sélectionner tous les formulaire présent dans la table en question. Je les affiche donc dans un container pour avoir de la lisibilité sur la page. La fonctionnalité de cette page est de supprimer le formulaire de contact. Par exemple si la demande a déjà pu être traité, l'employé ou l'admin aura juste à la supprimer pour laisser place à des demandes en attente de traitement. Cette page est accessible aux employés et à l'admin seulement.

CONTACT

Pour cette page, je viens récupérer tous mes formulaire de contact. Je les récupère avec une requête SQL qui va me servir à sélectionner tous les formulaire présent dans la table en question. Je les affiche donc dans un container pour avoir de la lisibilité sur la page. La fonctionnalité de cette page est de supprimer le formulaire de contact. Par exemple si la demande a déjà pu être traité, l'employé ou l'admin aura juste à la supprimer pour laisser place à des demandes en attente de traitement. Cette page est accessible aux employés et à l'admin seulement.

RÉPARATION

Dans cette page, il y a deux fonctionnalités. La première est pour modifier le bandeau de réparation, la deuxième est pour modifier la page réparation. Pour le bandeau, je récupère toute la table et je boucle dessus pour tout afficher dans un formulaire. On peut alors y modifier l'image principal ainsi que le titre. Toutes ces données seront stockés dans des variables et envoyé directement en base de donnée.

Pour la seconde fonctionnalité, c'est donc pour modifier les descriptions sur la page parlant des réparation. Je récupère toutes les descriptions et les titres, et l'admin peut alors tout changer comme bon lui semble. Je prépare la requête SQL en insérant ces données à l'intérieur pour que tout puisse être changé en base de donnée. Cette page est accessible à l'admin seulement.

INSCRIRE UN EMPLOYÉ

Pour cette page, j'ai récupéré mon script d'inscription. Tous les nouveaux utilisateur inscrit auront automatiquement le rôle « Employé ». Cette page est accessible à l'admin seulement.

MODIFIER UTILISATEUR

Pour cette page, je viens récupérer tous mes utilisateurs et je viens faire une boucle foreach en affichant dans un formulaire les données de chaque utilisateur, sauf le mot de passe. Tout est modifiable sauf l'ID du compte. Je récupère l'ID du compte lors que l'utilisateur clique sur le bouton pour appliquer ses modifications. J'inclue l'ID à la requête SQL pour changer les informations uniquement de l'utilisateur en question. Cette page est accessible uniquement par l'Admin.

SUPPRIMER UTILISATEUR

Pour cette page, je viens récupérer tous mes utilisateur et je viens faire une boucle foreach en affichant dans un formulaire les données de chaque utilisateur, sauf le mot de passe. Je récupère l'ID du compte lors que l'utilisateur clique sur le bouton pour supprimer l'utilisateur sélectionné. J'inclue l'ID à la requête SQL pour supprimer uniquement l'utilisateur en question. Cette page est accessible uniquement par l'Admin.

HÉBERGER EN LIGNE

Pour publier mon site en ligne, j'ai trouvé un hébergeur gratuit du nom de AlwaysData. Pour envoyer les fichiers directement sur le nom de domaine, j'ai du utiliser le protocole FTP, je me suis servi de FileZila pour upload tous mes fichiers sur mon site web.

EXPORT BASE DE DONNÉE

Pour exporter et importer ma base de donnée, j'ai déjà du l'exporter de mon site local vers ma base de donnée en ligne. En voulant l'importer, j'ai eu plusieurs erreurs, et après de longues recherches, j'ai remarqué que l'encodage n'était alors pas bon. J'ai donc changé à la main tout l'encodage de ma base de donnée en un format général.

LLIMOS FLORIAN