

UN BUREAU VIRTUEL DE GESTION DES LOCATIONS DE VOITURES



FANGYUAN YE Lisa - RATOVO Mélodie - GHARBI Roya

Groupe 203 - Groupe 204

Projet PWEB

Table des matières

Présentation du projet.....	2
Rappel des services déjà implémentés	3
Nouveaux services implémentés.....	4
Ajouter une voiture.....	4
Supprimer une voiture	5
Louer une voiture	6
Service de facturation	7
Crypter les mots de passe	9
Amélioration du design	9
Architecture du code	10
Schéma relationnel de la base	12
Schéma et explication	12
Information pour la connexion.....	13
Table sur PHPMyAdmin.....	14
Outils et langages utilisés	16
Conclusion	17

Présentation du Projet

Notre application permet à plusieurs entreprises de louer une flotte de véhicules. Ces véhicules sont mis à disposition grâce à un loueur, Berger. Lorsqu'on entre sur notre site, il y a une page d'accueil avec plusieurs véhicules disponibles pour la location. Si l'on veut en louer une, il faudra se connecter ou s'inscrire en tant qu'entreprise cliente. Si l'entreprise est déjà abonnée, elle pourra voir l'ensemble des véhicules qu'elle a déjà loués et la facture pour chaque voiture. Sinon, elle n'aura aucun véhicule sur sa page. Pour louer une voiture, il faut cliquer sur le bouton Louer Voitures et choisir la voiture souhaitée. Un formulaire apparaîtra pour louer cette voiture (mettre les dates...) . Le loueur peut effectuer plusieurs tâches. Il peut tout d'abord, ajouter ou retirer une ou plusieurs voiture(s) du stock. Il peut voir quelles sont les voitures en stock et quelles sont les voitures en cours de location (liste des entreprises clientes avec les voitures louées). Et enfin, il peut aussi avoir accès aux factures.

Rappel des services déjà implémentés

Service fonctionnant au moment de la pré-évaluation en fin octobre :

Au moment de la pré-évaluation, nous avons déjà plusieurs services fonctionnels. Tout d'abord l'accueil des utilisateurs non abonnés. Ils arrivent sur une page qui présente les différentes voitures que proposent le loueur et qui sont disponibles pour la location. Ensuite, on retrouve la partie dédiée aux entreprises clientes. Si l'entreprise a déjà un compte, elle peut directement se connecter. Sinon elle pourra s'inscrire via un formulaire d'inscription. Chaque entreprise cliente aura une page personnalisée avec ses données, sa flotte de voitures. Puis il y a aussi la partie destinée au loueur, il devra se connecter. Arrivé sur sa page, le loueur aura accès à la liste de véhicules qu'il a en stock et les véhicules qu'il a loués. Nous avons aussi commencé le service "ajouter une voiture" mais il n'était pas encore tout à fait fonctionnel. De même pour le service "supprimer une voiture".

Nouveaux services implémentés

- **Ajouter une voiture au stock**

Parmi les nouveaux services que nous avons réalisés , on retrouve "ajouter une voiture". Depuis sa page d'accueil, le loueur pourra inscrire de nouvelles voitures à sa flotte. Afin d'ajouter une nouvelle voiture, le loueur devra fournir plusieurs informations via un formulaire .Le type de véhicule (ex : Clio), le nombre de voiture qu'il veut ajouter, le prix par jour et les caractéristiques de la voiture (moteur, vitesse, nombre de places). Le loueur devra aussi importer l'image correspondante qui se fera avec un bouton. Si les données sont bien inscrites dans la base, la voiture sera ajoutée et un message le confirmera. Sinon, s'il y a eu une erreur, un message d'erreur s'affiche en indiquant l'erreur en question. Toutes les informations entrées par le loueur doivent respecter de nombreuses contraintes comme par exemple le nombre de véhicules et le prix doit forcément être un entier, il ne peut pas y avoir deux types de véhicules avec le même nom, il peut y avoir deux images identiques et d'autres encore.



The image shows a web form titled "Ajouter une voiture" on a yellow background. The form is contained within a rounded rectangle and includes the following elements:

- A text input field for "Type vehicule" with the placeholder text "Ex: Clio".
- A text input field for "Nb Vehicule" with the placeholder text "Ex: 5".
- A text input field for "Prix par jour" with the placeholder text "Ex: 13".
- Three text input fields for "caracteres:" with labels "- moteur", "- vitesse", and "- places". The first has the placeholder "Ex: hybride", the second "Ex: automatique", and the third "Ex: 5".
- A file upload section with a button labeled "Choisir un fichier" and the text "Aucun fichier choisi".
- A "Submit" button in the bottom right corner.

- **Supprimer une voiture au stock**

Pour la fonction supprimer, nous avons choisi de créer un formulaire. Sur la page du loueur, il y a un bouton qui permet de supprimer une ou plusieurs voitures du stock. Ce bouton nous renvoie à un formulaire où nous devons insérer le type du véhicule et le nombre de véhicules que nous voulons retirer du stock. Pour élaborer ce processus de suppression, il nous a fallu créer deux fonctions, l'une dans le fichier "operationLoueurBD.php" et l'autre dans "operationLoueur.php". Un fichier tpl était aussi nécessaire pour faire la mise en page. En fonction du nombre de véhicules que l'on veut retirer, le véhicule sera soit retiré de la base de données car le nombre de véhicules sera inférieur à 0, soit la colonne nombre de véhicules sera modifiée (grâce à une soustraction). Nous avons aussi mis des contraintes pour ne pas supprimer une voiture en cours de location.



The image shows a web form titled "Supprimer une voiture" (Delete a car) on a yellow background. The form is contained within a rounded rectangle and has two input fields. The first field is labeled "Type vehicule" and has a placeholder "Ex: Clio". The second field is labeled "Nombre de Véhicule" and has a placeholder "Ex: 5". A "Submit" button is located at the bottom right of the form.

Type vehicule
Ex: Clio

Nombre de Véhicule
Ex: 5

Submit





○ Louer une voiture

Un des autres services que notre application propose aussi est évidemment de louer une voiture. En arrivant sur la page d'accueil de l'application, les entreprises pourront voir les différentes voitures disponibles pour la location. Sous chaque type de voiture, on retrouve le bouton "Louer". Avant de pouvoir louer une voiture, il sera demandé à l'entreprise de se connecter ou de créer un compte. Après cela, l'utilisateur devra remplir un formulaire en indiquant obligatoirement la date de début de la location. La date de fin n'est pas obligatoire. L'utilisateur pourra choisir les dates à travers un calendrier. Certaines contraintes ont été ajoutées pour éviter des erreurs comme par exemple la date de début de la location doit être supérieure à la date du jour. Le client devra aussi inscrire le nombre de voitures qu'il souhaite louer et qui devra correspondre à l'ensemble de la flotte proposer par le loueur. Le client aura alors le choix de payer directement ou de payer plus tard. S'il n'y a pas eu d'erreur, la facture sera affichée. Sinon un message d'erreur s'affiche en indiquant l'erreur en question.

The image displays two screenshots of a car rental application. The top screenshot shows the home page with a banner for "Location de voiture" and four car models: Polo, Astra, 207, and Bmw. Each car listing includes its name, availability, price per day, engine type, transmission, and number of seats, along with a "Louer" button. The bottom screenshot shows the "Louer Voiture" form, which includes fields for "Date début", "Date fin", and "Nb Vehicule", and buttons for "Payer Plus tard" and "Payer Directe".

Home Louer Entreprise

Location de voiture

Polo	Astra	207	Bmw
Nombre de véhicule disponible : 6 Prix/jour : 20	Nombre de véhicule disponible : 2 Prix/jour : 13	Nombre de véhicule disponible : 6 Prix/jour : 16	Nombre de véhicule disponible : 6 Prix/jour : 26
			
Moteur : diesel Vitesse : automatique Places : 5	Moteur : diesel Vitesse : automatique Places : 5	Moteur : essence Vitesse : manuelle Places : 5	Moteur : diesel Vitesse : automatique Places : 5
Louer	Louer	Louer	Louer

Home Retourne

Louer Voiture

Date début:

Date fin:

Nb Vehicule:

Payer Plus tard Payer Directe

© 2020


○ Service de facturation

On propose aussi le service de facturation. L'entreprise cliente vient de louer une voiture. S'il n'y a pas d'erreur, la facture s'affiche à la fin de l'opération. On trouve sur la facture le numéro de la facture, le nom du client et un tableau récapitulatif des informations de la voiture. Mais les factures peuvent diverger en fonction des choix du client. Le client peut choisir de régler directement ou pas. L'état du paiement sera indiqué sur la facture ainsi que le montant non payé. Pour calculer le nombre de jours de location, on effectue une soustraction entre date fin et date début. Pour effectuer cette soustraction, on convertit les deux dates en estampilles et on évalue la durée les séparant $((dateDebut - dateFin) / 86400) + 1$. Si le client a loué la voiture pendant plus d'un mois, il trouvera sur la facture le montant du mois courant. On obtient le montant du mois avec le calcul : $30 * nb \text{ voiture} * \text{prix} / j$. Le client pourra bénéficier d'une réduction de 10% s'il loue une flotte de 10 voitures. Les clients ou le loueur pourront accéder à tout moment à ses factures avec un bouton sur sa page d'accueil personnelle.

[Home](#)[Accueil](#)

Facture n°11

Client : Pinson



Voiture louée depuis le 2020-06-28.
Prix de la voiture "Clio" par jour est de 12 euros.
Vous avez loué 1 véhicules pendant 179 jours.
Le montant du mois courant est de 1800 euros

Etat de paiement : Payé

Recapitulatif :

Qte	Description	Prix /j	Montant
1	Id : 1 Type : Clio Date début : 2020-06-28 Date fin : 2020-12-22	12€	1800€
		Total :	1800€


- Service de facturation

Le service de facturation est le même que pour le client. Le loueur peut quant à lui avoir une liste de toutes les factures des clients. Il peut voir les montant des factures du mois courant. Le loueur à la possibilité de voir le statut de chaque facture si elles sont payés ou pas.

[Home](#)[Accueil](#)

Facturation de Client n°1

Client : Suly



Recapitulatif :

Qté	Description	Prix /j	Montant (de ce mois)	Paiement
5	<small>Id Vehicule: 10 Type : 208 Date début : 2020-08-01 Date fin : 2022-06-01</small>	14€	530€	Payé
7	<small>Id Vehicule: 4 Type : Golf Date début : 2019-10-01 Date fin : 2022-06-30</small>	10€	500€	Payé
Montant total		Montant payé	Montant non payé	
1030		1030	0	

© 2020

- **Crypter les mots de passe**

Nous avons utilisé la fonction : " crypt (string \$str [, string \$salt]) : string " pour sécuriser les mots de passe du client. En effet, la fonction crypt() va chiffrée le mot de passe à sens unique selon le paramètre « salt » qu'on lui fournit. Nous avons décidé d'utiliser le hachage CRYPT_MD5. Il va chiffrer les mots de passe en 12 caractères commençant par \$1 \$. En plus, comme la fonction crypt() va chiffrée à sens unique, il est indéchiffrable. Pour que les clients puissent se connecter, on crypte aussi le mot de passe qu'il vient d'entrer. Ensuite on compare ces mots de passe pour déterminer si ce client existe ou pas.

- **Amélioration du design**

Pour avoir une interface graphique beaucoup plus agréable, nous avons choisi de modifier la mise en page de tous les services. Tout d'abord, nous avons choisi les couleurs pour l'ensemble du site. Nous sommes parties sur trois couleurs à savoir le blanc, le gris mat et l'or métallisé. Ensuite, nous avons choisi plusieurs typographies : Arial Black, Crimson Text, Arial, Helvetica, et Sansita Swashed. On trouve une barre de navigation qui permet de circuler facilement dans l'ensemble de l'application. Sur la page d'accueil, on retrouve seulement les éléments importants : une image large, la fonction de notre application, et les voitures disponibles pour la location. Les voitures sont disposées un par un dans différents blocs. Toutes les pages avec un formulaire ont le même design. Le titre en blanc, la couleur de fond, un dégradé d'or métallisé et les contours en gris mat. Pour les pages d'accueil loueur et clients abonnés, les textes sont en gris mat. Nous avons choisi d'afficher les flottes de voiture dans des tableaux.

Architecture du code

Pour architecturer notre application web, nous avons suivi le modèle MVC. Nous avons donc créé 3 parties, Vue, Modèle, et Contrôle. Cette décomposition nous permet de séparer le code de l'interface (fichiers tpl) , le code permettant d'accéder à la base de données (fichiers php avec les fonctions avec des requêtes sql + fichier sql avec les créations des tables et les insert) et la partie du code qui gère la dynamique de l'application c'est-à-dire le code qui permet d'avoir un contenu personnalisé pour chaque utilisateur. Ces deux dernières parties contiennent des fichiers PHP. On peut alors savoir clairement dans quelle partie accéder si l'on veut effectuer une manipulation précise (modifier l'interface utilisateur...). Seule la partie Vue est séparée en plusieurs dossiers pour nous permettre de savoir dans quel service nous nous situons. Par exemple, si nous voulons modifier l'interface de la page d'accueil des entreprises, il suffit d'accéder à la partie Vue puis à la partie entreprise et enfin au fichier « accueilEntrep.tpl ». Chaque service possède un dossier à son nom pour faciliter la recherche.

La partie Contrôle possède les fichiers php avec des fonctions qui appellent les autres fichiers contenus dans les autres parties. On utilise pour cela « require » et on insère le chemin pour accéder au fichier souhaité. Par exemple, dans le fichier loueur.php qui est dans la partie Contrôle, il y a une fonction « connexionLoueur() ». Dans cette fonction, nous appelons le fichier « connexionLoueur.tpl » pour avoir l'interface graphique mais aussi le fichier « loueurBD.php » pour nous permettre d'accéder à la base en fonction de la requête définie dans celle-ci.

Le fichier connect.php nous permet d'accéder à la base c'est-à-dire d'ouvrir une connexion à un serveur. Ici elle se fera sur PHPMyAdmin. Le fichier index.php est l'entrée de notre site. On y a mis la première page que l'on doit avoir lorsqu'on ouvre notre site.

Chaque fichier a pour nom le service dans lequel il effectue des modifications comme facturation, loueur, entreprise et plus encore. Voici dans la page suivante comment sont répartis les différents fichiers.

Architecture du code

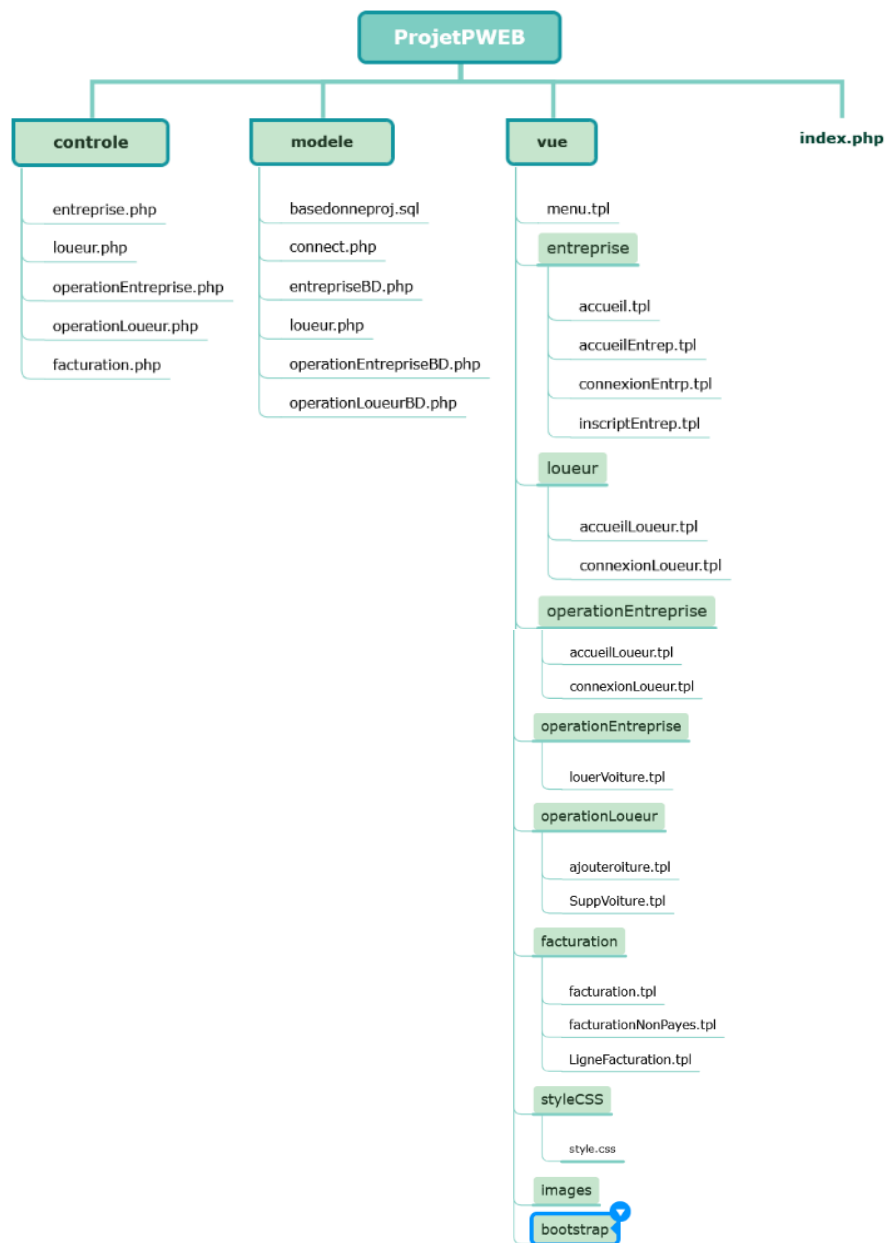
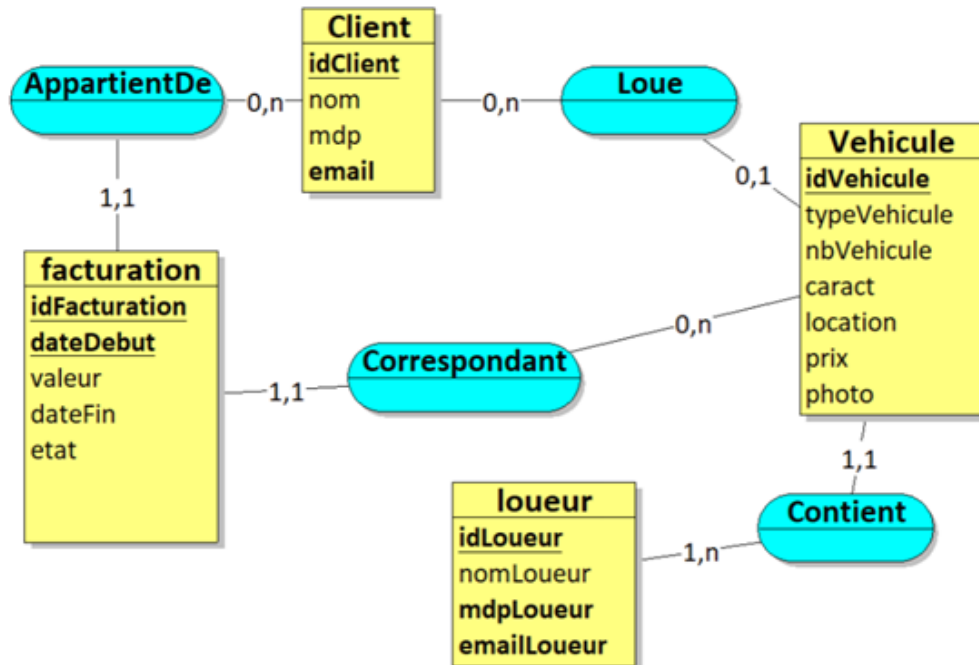


Schéma relationnel de la base

○ Schéma et explication



Le chemin pour accéder à la base de données :

\\UwAmp\\www\\ProjetPWEB\\modele\\basedonneproj.sql

Notre base de données n'a pas énormément changé par rapport au premier rendu. Nous avons toujours nos tables loueur, facturation, client et véhicule. Cette fois-ci, la table facturation est fonctionnelle. Elle nous permet d'avoir des factures en fonction des véhicules. Nous avons toujours qu'un seul loueur avec des coordonnées uniques. Les clients peuvent être ajoutés dans la base en insérant leurs données personnelles à travers un formulaire d'inscription. Quant à la table Véhicule, nous avons ajouté une colonne qui est le « prix ». Nous n'avons pas pensé à cette colonne au début du projet car le service facturation n'était pas fonctionnel. Mais après l'avoir fait, nous nous sommes rendu compte qu'il nous manque un champ pour compléter la relation entre les deux tables et réaliser la facture correspondant à un véhicule. Les noms et les mots de passe n'ont pas changé que ce soit pour le client ou pour le loueur.

- **Information pour la connexion**

Loueur :

- Nom: Berger
- Mot de passe: 74U6YidAt

Clients :

Client 1 :

- Nom: Suly
- Mot de passe: LLphQk673

Client 2:

- Nom: Pinson
- Mot de passe: 8Vjnz2JB9

Client 3:

- Nom: Emery
- Mot de passe: hU7bK3d2M

Client 4:

- Nom: Daire
- Mot de passe: 47MMGt4ik

Client 5:

- Nom: Yart
- Mot de passe: W6yX4gHh7

○ Tables sur PHPMyAdmin

Les 4 tables (loueur, client, vehicule, facturation) :

Table	Action	Lignes	Type	Interclassement	Taille	Perte
<input type="checkbox"/> client	★ Afficher Structure Rechercher Insérer Vider Supprimer	5	InnoDB	utf8_bin	16 Kio	-
<input type="checkbox"/> facturation	★ Afficher Structure Rechercher Insérer Vider Supprimer	12	InnoDB	utf8_bin	48 Kio	-
<input type="checkbox"/> loueur	★ Afficher Structure Rechercher Insérer Vider Supprimer	1	InnoDB	utf8_bin	48 Kio	-
<input type="checkbox"/> vehicule	★ Afficher Structure Rechercher Insérer Vider Supprimer	10	InnoDB	utf8_bin	48 Kio	-
4 tables	Somme	28	InnoDB	utf8_bin	160 Kio	0

Table loueur :

		idLoueur	nomLoueur	mdpLoueur	emailLoueur
<input type="checkbox"/>	Modifier Copier Effacer	1	Berger	74U6YidAt	jberger@free.fr

Table client :

		idClient	nom	mdp	email
<input type="checkbox"/>	Modifier Copier Effacer	1	Suly	e7yabzB1g5lnk	suly@free.fr
<input type="checkbox"/>	Modifier Copier Effacer	2	Pinson	6f.GBojfb9XWA	pinson.martha@yahoo.fr
<input type="checkbox"/>	Modifier Copier Effacer	3	Emery	2coXXuq2ZDJT.	emery@gmail.com
<input type="checkbox"/>	Modifier Copier Effacer	4	Daire	2akkshins72fA	pauletaire@gmail.com
<input type="checkbox"/>	Modifier Copier Effacer	5	Yart	d7VnhQweQY.hY	m.yart@free.fr

Table vehicule :


































		idVehicule	typeVehicule	nbVehicule	caract	location	prix	photo	idLoueur	idClient
<input type="checkbox"/>	  	1	Clio	5	{"moteur":"hybride","vitesse":"automatique","place...	non_disponible	12	[BLOB - 30 o]	1	2
<input type="checkbox"/>	  	2	Megane	3	{"moteur":"electrique","vitesse":"automatique","pl...	non_disponible	14	[BLOB - 32 o]	1	5
<input type="checkbox"/>	  	3	Twingo	2	{"moteur":"electrique","vitesse":"automatique","pl...	non_disponible	11	[BLOB - 32 o]	1	5
<input type="checkbox"/>	  	4	Golf	7	{"moteur":"diesel","vitesse":"manuelle","places":5...	non_disponible	10	[BLOB - 30 o]	1	1
<input type="checkbox"/>	  	5	Polo	6	{"moteur":"diesel","vitesse":"automatique","places...	disponible	20	[BLOB - 30 o]	1	NULL
<input type="checkbox"/>	  	6	Corsa	3	{"moteur":"diesel","vitesse":"automatique","places...	non_disponible	17	[BLOB - 31 o]	1	2
<input type="checkbox"/>	  	7	Astra	2	{"moteur":"diesel","vitesse":"automatique","places...	disponible	13	[BLOB - 31 o]	1	NULL
<input type="checkbox"/>	  	8	207	6	{"moteur":"essence","vitesse":"manuelle","places":...	disponible	16	[BLOB - 29 o]	1	NULL
<input type="checkbox"/>	  	9	Fiesta	2	{"moteur":"electrique","vitesse":"automatique","pl...	non_disponible	15	[BLOB - 32 o]	1	3
<input type="checkbox"/>	  	10	208	5	{"moteur":"diesel","vitesse":"automatique","places...	non_disponible	14	[BLOB - 29 o]	1	1

Table facturation :

		idFacturation	dateDebut	valeur	dateFin	etat	idVehicule	idClient
<input type="checkbox"/>	  	1	2020-02-26	900	2020-06-26	1	9	3
<input type="checkbox"/>	  	2	2019-12-15	2100	2020-05-01	1	10	1
<input type="checkbox"/>	  	3	2020-08-07	1260	2020-11-15	0	2	5
<input type="checkbox"/>	  	4	2020-08-01	2100	2022-06-01	1	10	1
<input type="checkbox"/>	  	5	2018-08-15	780	0000-00-00	1	7	5
<input type="checkbox"/>	  	6	2020-03-22	592	2020-06-26	1	1	3
<input type="checkbox"/>	  	7	2019-10-05	1530	2020-12-01	1	6	2
<input type="checkbox"/>	  	8	2020-01-16	457	2020-05-25	1	8	4
<input type="checkbox"/>	  	9	2019-10-01	2100	2022-06-30	1	4	1
<input type="checkbox"/>	  	10	2020-08-05	660	NULL	0	3	5
<input type="checkbox"/>	  	11	2020-06-28	1800	2020-12-22	1	1	2
<input type="checkbox"/>	  	12	2020-08-15	900	2021-01-10	0	9	3

Outils et langages utilisés

Cette application a été faite à l'aide de plusieurs langages informatiques. Nous avons utilisé les langages de bases pour l'interface graphique qui sont les langages HTML et CSS. Celles-ci nous ont permis de créer les différentes pages pour les différents types de service. Pour avoir un contenu personnalisé en fonction des utilisateurs, nous avons utilisé PHP. Nous avons stocké l'ensemble des données codées en SQL dans PHPMyAdmin grâce au serveur UwAmp. À partir des fichiers PHP, nous pouvions alors extraire les données souhaitées de la base et les utiliser en fonction de nos besoins. La création des codes a été faite soit sur Notepad++ ou sur Visual code.



Conclusion



Ce projet nous a permis de développer nos connaissances en PHP mais aussi en SQL, HTML et CSS. Nous savons désormais créer une application à l'aide de ces différents langages et répondre aux besoins des utilisateurs. Malgré de nombreuses difficultés nous avons réussi à faire le travail demandé. Nous avons réussi à créer différents services pour différents types d'utilisateurs avec des besoins plus ou moins similaires.

Certains services ne sont pas parfaits comme le service de facturation, la suppression des voitures n'est pas parfaite. Dans la page d'accueil, il est préférable de faire un panier pour sélectionner les voitures que l'on souhaite louer.

Comme une application n'est jamais parfaite, il faut sans cesse l'améliorer en utilisant de nouvelles syntaxes, de nouvelles fonctions. C'est pourquoi, nous avons encore beaucoup de choses à apprendre pour avoir la meilleure version possible de notre application. Nous pouvons donc encore plus améliorer notre code pour sécuriser davantage la base de données et avoir un code sans redondance et clair.