



Dossier de projet

Création d'un site vitrine

www.tbautos.fr

Titre professionnel

Développeur web et web mobile

Stage effectué chez TB AUTOS

Du 01/03/2021 au 26/04/2021

Formation Access Code School

Es sabbani Nasser



Sommaire

○ Remerciements	Page 3
○ Présentation et objectifs	Page 4
○ Technologies utilisées pour le développement	Page 6
○ Liste des compétences du référentiel	Page 7
○ Cahier des charges	Page 8
○ Réalisations comportant les extraits de code	
○ I - Header / Footer	Page 9
○ II - Page d'accueil	Page 10
○ III - Carte grise	Page 14
○ IV - Véhicule / Détail des véhicules	Page 15
○ V - Réparation (Centre automobile)	Page 17
○ VI - Contact	Page 20
○ VII - Prise de rendez-vous	Page 22
○ Réalisation d'une administration	page 25
○ Présentation du jeu d'essai de la fonctionnalité la plus représentative	Page 34
○ Veille effectuée durant le projet sur les vulnérabilités de sécurité	Page 36
○ Situation de travail ayant nécessité une recherche à partir de site anglophone	page 38
○ Extrait du site anglophone et traduction	Page 39
○ Annexes	Page 40



Remerciements

Je remercie Monsieur Tbatou Karim tout d'abord pour son accueil au sein de sa société **TB AUTOS** mais aussi pour tout ce qu'il a pu m'apporter pendant ces 2 mois de stage à ses côtés. Ses conseils avisés m'ont permis d'apprendre énormément de choses et donc d'acquérir de nouvelles connaissances et compétences. Je le remercie également pour la confiance qu'il a su m'accorder dès les premiers instants.

Je remercie également l'ensemble des personnes avec qui j'ai pu travailler, et mon tuteur de stage Alain Merucci. Ils ont su se rendre disponibles quand cela été nécessaire et ont toujours pris soin de m'expliquer les choses de façon pédagogique.

Merci à toutes ces personnes qui ont fait de mon stage une réussite.



Présentation de la société

L'entreprise **TB AUTOS** possède un bureau de carte grise situé au [65 Rue du Collège, 39200 Saint-Claude](#), où nous avons la possibilité d'effectuer un changement de carte grise car en effet, depuis 2017, les préfectures et sous-préfectures n'accueillent plus le public pour le changement des cartes grises.



La société **TB AUTOS** possède un centre automobile situé à la [Zone Industrielle du Curtillet, 39170 Pratz](#). Le garage effectue les réparations et l'entretien de votre véhicule. **TB AUTOS** offre un service de qualité pour tous les types de véhicules. Tous les services mécaniques sont effectués par des mécaniciens qualifiés et offre une gamme complète de services de garage pour répondre à toutes demandes clientèle.





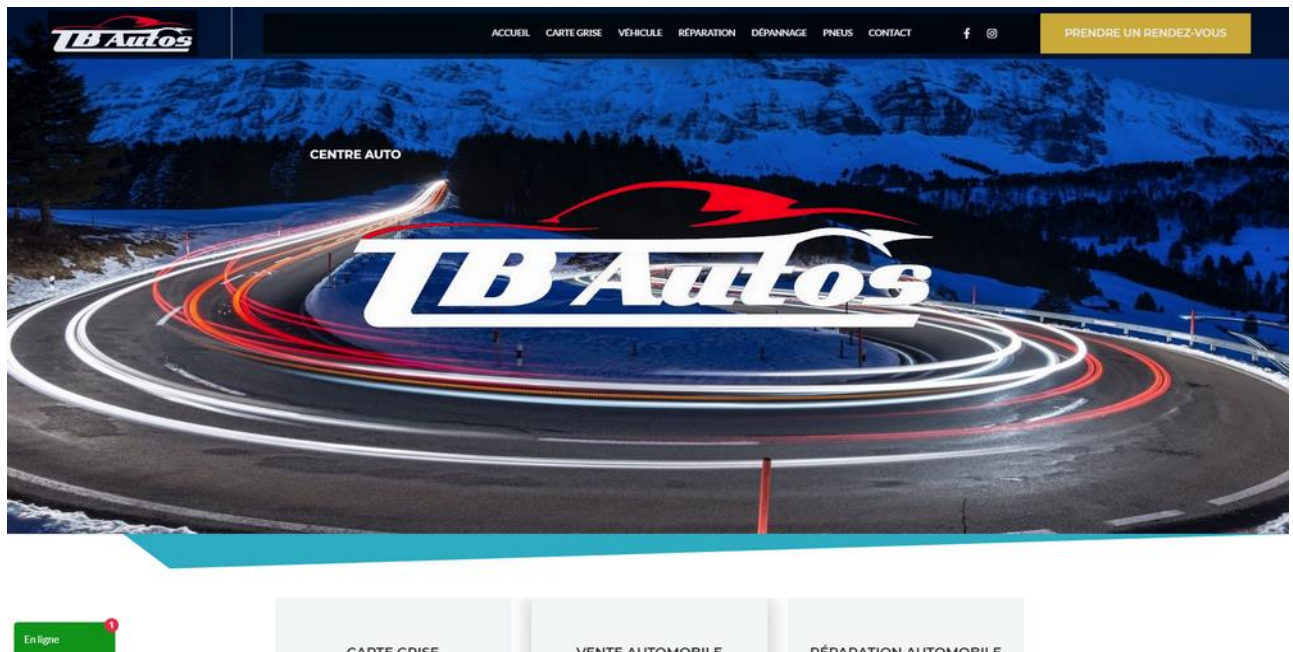
Objectifs

Mon objectif principal est la réalisation d'un site vitrine pour la société **TB AUTOS**.

Un **site vitrine** est un **site internet** qui présente les produits et services d'une activité, sans vendre en ligne. Un **site vitrine** a donc **pour objectif** d'apporter une présence en ligne à une entreprise, un artisan, une collectivité et autres, afin d'acquérir des contacts, prospects et clients via **internet**.

Les objectifs définis avant la création du site internet :

- Acquisition de nouveaux clients / prospects
- Amélioration de la notoriété et l'image de la société
- Information au sujet de l'entreprise (Carte grise et centre automobile).
- Promouvoir le service de carte grise
- Promouvoir le centre automobile (Réparation et vente de véhicule)
- Service client (Répondre en ligne selon la demande du client)
- Réalisation d'une administration (Gérer les avis, prise de rendez-vous ...)
- Prise de rendez-vous





Technologies utilisées pour le développement

Wamp :

WampServer est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans avoir à se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant trois serveurs (Apache, MySQL et MariaDB), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL.

Pour réaliser le projet, j'ai commencé par installer wamp pour pouvoir travailler en local.

Jquery :

jQuery est une bibliothèque JavaScript libre et multiplateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web². La première version est lancée en janvier 2006 par John Resig.

Javascript :

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs avec l'utilisation (par exemple) de Node.js

J'ai utilisé Javascript et Jquery pour rendre le site dynamique.

Photoshop :

Photoshop est un logiciel de retouche, de traitement et de dessin assisté par ordinateur, lancé en 1990 sur MacOS puis en 1992 sur Windows.

Photoshop a été utilisé dans ce projet pour retravailler certaines images du site.

Bootstrap :

Bootstrap est une collection d'outils utiles 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.





Liste des compétences du référentiel couvertes par le projet

Développer la partie frontend d'une application web ou web mobile :

Pour la réalisation du projet, il a été nécessaire de :

Créer une interface utilisateur web statique et adaptable :

- Les pages web s'adaptent à la taille de l'écran
- Le site respecte les règles de référencement naturel

Développer une interface web dynamique :

- Les pages web sont conformes à l'expérience utilisateur y compris pour l'expérience mobile
- L'application web est optimisée pour les équipements mobiles

Développer la partie backend d'une application web ou web mobile en intégrant les recommandations de sécurité :

Développer la partie backend d'une application web ou web mobile :

- Concevoir et utiliser une base de données MySQL
- Développer les composants d'accès aux données
- Identifier et comprendre les besoins clients et utilisateurs





Cahier des charges

Contexte :

Comme expliqué plus haut, l'objectif du projet est la réalisation d'un site vitrine qui permettra à la société de gagner en notoriété, de viser plus de nouveaux clients, et de présenter les différents services de la société.

Conditions pour la réalisation du projet :

Logo :



Header :

- Logo présent dans le header
- Affichage des différentes pages du site.

Template :

- Nous avons choisi une template Bootstrap de base
Source de la Template (Template car autlines)
- Nous avons fait un point sur la modification de la Template.

Footer :

- Logo présent dans le footer
- Affichage des différentes pages du site.
- Affichage des informations de la société.

Sur le site en général :

- Adaptable en version mobile
- Les clients peuvent interagir (En ajoutant des avis, ou grâce à un live chat).

C'est donc avec ces consignes que j'ai commencé à intégrer mes idées.



Réalisation

Pour la réalisation du site vitrine en backend, j'ai travaillé en procédural à l'aide d'un fichier que j'ai nommé « fonction.php ».

Ensuite, lorsque j'ai besoin d'une fonction, il me suffit d'appeler la fonction sur une des pages du site vitrine, comme par exemple la réalisation du système d'avis à la page d'accueil, que j'expliquerais par la suite.

I – Header / Footer :

Tout d'abord, j'ai procédé par la création de deux fichiers PHP :

- header.php,
- footer.php

Par la suite, j'appelle ces deux fichiers PHP à l'aide d'une requête, afin que lorsque je fais une modification sur l'une des deux pages, cela s'affiche sur l'ensemble des pages du site vitrine :

```
<?php include('footer.php'); ?>
```

```
<?php include('header.php'); ?>
```



II – Page d'accueil :

Pour la réalisation de la page d'accueil, j'ai travaillé en priorité sur le système d'avis, afin d'apporter une meilleure notoriété et image à l'entreprise.

Pour commencer le slider (système d'affichage d'avis) était déjà présent dans la template,

1 - J'ai donc tout d'abord, créé une table dans la base de données sous le nom 'avis' :

Serveur: MySQL:3306 » Base de données: thautos » Table: avis									
Parcourir Structure SQL Rechercher Insérer Exporter Importer Privilèges Opérations Déclencheurs									
#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/> 1	id	int			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
<input type="checkbox"/> 2	description_avis	varchar(255)	latin1_swedish_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/> 3	nom_avis	varchar(255)	latin1_swedish_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/> 4	rate	varchar(255)	latin1_swedish_ci		Non	Aucun(e)			Modifier Supprimer Plus

description_avis = Commentaire.

nom_avis = Nom Prénom, de la personne qui a publié le commentaire.

Rate = Note du client sur l'entreprise (système d'étoile)

2 – Mon maître de stage m'a demandé d'afficher seulement les 5 derniers avis, j'ai donc créé une fonction, qui permettra d'afficher les informations de la table avis.

```
function avis(){
    $bdd=$bdd();
    $recuperation = $bdd->prepare('SELECT * FROM avis ORDER BY id DESC LIMIT 5');
    $recuperation->execute();
    while ($tache = $recuperation->fetch()) {
        echo '
        <blockquote class="b-reviews">
        <div class="b-reviews__text">'.$tache['description_avis'].'</div>
        <div class="b-reviews__footer">
        <center><br><br></center>
        <div class="b-reviews__name">'.$tache['nom_avis'].'</div>
        </div>
        </blockquote>
        ';
    }
}
```



Je crée une connexion à la base de données « \$bdd=bdd(); ».

Je récupère les informations de la table 'avis' à l'aide de la requête ' SELECT * FROM 'avis' j'affiche les 5 derniers avis grâce à la requête, « ORDER BY id DESC LIMIT 5 » (sélectionne les 5 'id' en ordre décroissant).

Je récupère les informations de la base de données dans le cadre :

```
".$tache['description_avis']." ".$tache['rate']." ".$tache['nom_avis'].""
```

Résultat Frontend :



3 – Je crée un formulaire pour ajouter un avis :

Ajouter un avis

Commentaire :

Nom Prénom :



AJOUTER



Les avis s'ajoutent grâce à un formulaire qui communique avec une fonction :

```
<form name="ajout" method="POST">
  <div class="form-group">
    <div class="form-group">
      <label for="date_intervention">Commentaire :</label>
      <input type="text" id="description_avis" name="description_avis" class="form-control">
    </div>
    <label for="type_intervention">Nom Prénom :</label>
    <input type="text" id="nom_avis" name="nom_avis" class="form-control">
  </div>
  <div class="d-flex justify-content-center">
    <div class="form-group">
      <div class="rate">
        <div name="rate" class="star-rating">
          <input type="radio" id="5-stars" name="rate" value="avisimg/5etoiles.png" />
          <label for="5-stars" class="star">&bigstar;</label>
          <input type="radio" id="4-stars" name="rate" value="avisimg/4etoiles.png" />
          <label for="4-stars" class="star">&bigstar;</label>
          <input type="radio" id="3-stars" name="rate" value="avisimg/3etoiles.png" />
          <label for="3-stars" class="star">&bigstar;</label>
          <input type="radio" id="2-stars" name="rate" value="avisimg/2etoiles.png" />
          <label for="2-stars" class="star">&bigstar;</label>
          <input type="radio" id="1-star" name="rate" value="avisimg/1etoiles.png" />
          <label for="1-star" class="star">&bigstar;</label>
        </div>
      </div>
    </div><br><br>
    <div class="d-flex justify-content-center">
      <button class="btn btn-primary" type="submit" name="ad" value="OK">Ajouter</button>
    </div>
  </form>
  <?php addavis(); ?>
```

Je vérifie que le formulaire comporte les bonnes informations, afin que la fonction « addavis() ; » puisse communiquer avec le formulaire qui utilise la méthode « POST ».

Commentaire, je lui donne le nom de 'description_avis'

Nom Prénom, je lui donne le nom de 'nom_avis '

Pour la note, j'ai créé un dossier avec 5 images (1, 2, 3, 4, 5 étoiles), puis je donne le nom 'rate' et une valeur pour chaque image qui permettra d'avoir le chemin d'accès à l'image et d'afficher la note du client.

Bouton 'ajouter', je lui donne le nom de 'ad'.



Pour communiquer avec la fonction « addavis() », je dois tout d'abord appeler le fichier fonction.php puis ensuite la fonction « addavis() ; » :

```
<?php include('function.php'); ?>
```

```
<?php addavis(); ?>
```

Je vérifie par la suite les informations de la fonction « addavis() ; » :

```
function addavis(){
    $bdd=$bdd();
    if(isset($_POST['ad'])&&!empty($_POST['description_avis'])&&!empty($_POST['nom_avis'])&&!empty($_POST['rate'])) {
        $req = $bdd->prepare('INSERT INTO avis (id, description_avis, nom_avis, rate) VALUES (NULL, :description_avis, :nom_avis, :rate)');
        $req->execute(array(
            'description_avis' => $_POST['description_avis'],
            'nom_avis' => $_POST['nom_avis'],
            'rate' => $_POST['rate'],));

        echo 'La nouvelle tâche a été ajouté';
        echo '<script language="Javascript">
        document.location.replace("index.php");
        </script>';
    }
}
```

Les informations de ma requête SQL correspondent bien avec les informations du formulaire, je peux donc effectuer une insertion des informations à la base de données, une fois que j'appuie sur le bouton ajouter si tout se passe correctement, un message va s'afficher :

' La nouvelle tâche a été ajouté '

Et la page va être rafraîchie grâce au script en JavaScript, afin de voir les avis ajoutés.



III – Carte grise :

Pour la rubrique carte grise, j'ai utilisé principalement Bootstrap pour la mise en page, c'est à dire le texte à gauche et l'image à droite, et inversement pour la ligne du bas.

```
<div class="container">
<div class="row">
<div class="col-xs-12 col-md-6">
<br>
<p> Vous avez la possibilité d'effectuer votre changement de carte grise dans un de nos point de service <b>TB AUTOS</b>.<br><br>
En effet, depuis 2017, les préfectures et sous-préfectures n'accueillent plus le public pour le changement des cartes grises.<br><br>
Nous avons 2 points de service disponible : <br><br>
- <a href="https://www.google.fr/maps/place/65+Rue+du+Coll%C3%A8ge,+39200+Saint-Claude/@46.390744,5.8627003,17z/data=!3m1!4m5!3m4!1s3m4!1sd6jxOAVQwzbVVR1LwC4TF6Q!2e0!7i13312!8i6656" target="_blank">https://www.google.fr/maps/place/65+Rue+du+Coll%C3%A8ge,+39200+Saint-Claude/@46.390744,5.8627003,17z/data=!3m1!4m5!3m4!1s3m4!1sd6jxOAVQwzbVVR1LwC4TF6Q!2e0!7i13312!8i6656</a>
- <a href="https://www.google.fr/maps/@46.3841369,5.7708842,3a,75y,90.86h,78.07t/data=!3m6!1e1!3m4!1sd6jxOAVQwzbVVR1LwC4TF6Q!2e0!7i13312!8i6656" target="_blank">https://www.google.fr/maps/@46.3841369,5.7708842,3a,75y,90.86h,78.07t/data=!3m6!1e1!3m4!1sd6jxOAVQwzbVVR1LwC4TF6Q!2e0!7i13312!8i6656</a>
</div>
<div class="col-lg-6">

</div>
</div>
```

De plus, la page est responsive la class « img-fluid » permet de redimensionner l'image selon la taille de l'écran.

Pour les points de services de carte grise, j'ai fait en sorte pour que lorsque le client appuie sur l'adresse postale de l'un des points de service, une nouvelle page s'ouvre avec l'itinéraire pour se rendre dans l'un des points de service.

Exemple :

```
<a href="https://www.google.fr/maps/@46.3841369,5.7708842,3a,75y,90.86h,78.07t/data=!3m6!1e1!3m4!1sd6jxOAVQwzbVVR1LwC4TF6Q!2e0!7i13312!8i6656" target="_blank">Zone Industrielle du Curtillet, 39170 Pratz</a>
```

Href = Lien de référence pour se rendre au point de service.

<a> = Lien cliquable.

target="_blank" : Permet d'ouvrir la carte sur une autre page sans quitter le site.



IV – Véhicule / Détail des véhicules :

La page véhicule, permettra à l'utilisateur de voir les véhicules de la société, pour cela, je me suis inspiré du site : « leboncoin.fr ».

1- Tout d'abord, j'ai créé deux fichiers php :

- voiture.php / detail.php

2- Ensuite, j'ai créé une table dans la base de données sous le nom 'voitures' :

#	Nom	Type	Interclassement
1	id_voitures	bigint	
2	image_voitures	varchar(255)	latin1_swedish_ci
3	image_two	varchar(255)	latin1_swedish_ci
4	image_three	varchar(255)	latin1_swedish_ci
5	image_four	varchar(255)	latin1_swedish_ci
6	image_five	varchar(255)	latin1_swedish_ci
7	nom_voitures	varchar(255)	latin1_swedish_ci
8	prix_voitures	varchar(255)	latin1_swedish_ci
9	km_voitures	varchar(255)	latin1_swedish_ci
10	carburant_voitures	varchar(255)	latin1_swedish_ci
11	annee_voitures	varchar(255)	latin1_swedish_ci
12	boite_voitures	varchar(255)	latin1_swedish_ci
13	description_voitures	varchar(255)	latin1_swedish_ci

Les tables description_voitures, image_two, image_three, image_four, image_five n'apparaîtront pas dans la page voiture.php, mais apparaîtront dans la page detail.php.

Comme expliqué précédemment (Page 11), j'utilise la fonction fetch() ; afin de faire apparaître les informations de la base de données dans la page voiture.php.

3 – Chaque id représente un véhicule, l'objectif est de relier la page voiture.php avec la page detail.php, c'est-à-dire lorsque je clique sur l'image d'un véhicule ou sur le nom d'un véhicule, il faut que les détails du véhicule s'affichent.

```
<a href="detail.php?id='.$tache['id_voitures'].'"></a>
<a href="detail.php?id='.$tache['id_voitures'].'">'.$tache['nom_voitures'].'</a>
```



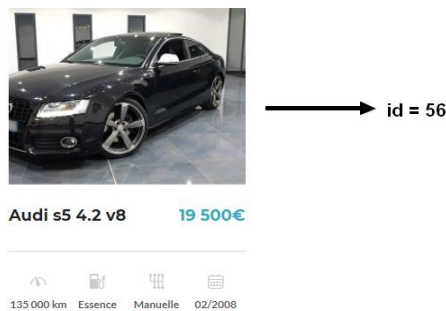
4 – Je récupère l'id du véhicule en passant par l'url :

```
<a href="detail.php?id='.$tache['id_voitures'].'">
```

5 – Une fois que les détails du véhicule s'affichent, je peux voir d'autres images grâce aux tables : image_two, image_three, image_four, image_five, et toutes les autres informations du véhicule.

Exemple :

Je clique sur l'image ou le titre de l'id 56 :



Je récupère dans l'url l'id 56 :

<https://tbautos.fr/detail.php?id=56>

Tous les détails du véhicule s'affichent grâce à l'id récupéré dans l'url et la fonction « fetch() ; » :





Pour finir, dans l'onglet voiture.php, j'ai créé une fonction permettant d'afficher le nombre de véhicules disponibles :

```
function compter() {  
    $bdd=bdd();  
    $recuperation = $bdd->prepare('SELECT count(*) as id from voitures');  
    $recuperation->execute();  
    if ($tache = $recuperation->fetch()) {  
        echo $tache['id'];  
    }  
}
```

La fonction sélectionne et compte tous les id présent dans la table 'voitures'.

Il y a actuellement 2 véhicules disponible

V – Réparation :

La page réparation, comporte des renseignements concernant l'activité du centre automobile de l'entreprise, pour cela mon maître de stage m'a demandé de faire un cadre avec une barre défilante, ou l'on pourra voir les 10 derniers avis des clients sur le centre automobile.

Pour cela, j'aurai besoin :

- Base de données
- Formulaire qui permettra d'écrire un avis
- La fonction fetch() ; qui permettra d'afficher les avis.

Tout d'abord, j'ai commencé par modéliser le cadre :

Nom prénom du client sera affiché en rouge

Avis du client sera affiché en noir.

```
<marquee direction="left" behavior="scroll" scrollamount="6" class="dedis" ><p class="indent"> <?php dedicace(); ?> </p></marquee>
```

La balise html <marquee> </marquee>, permet de faire défiler le texte.

direction="left" , permet de faire défiler le texte par la gauche.

L'attribut HTML **SCROLLAMOUNT** de la balise HTML MARQUEE définit de combien de pixels le texte avance.



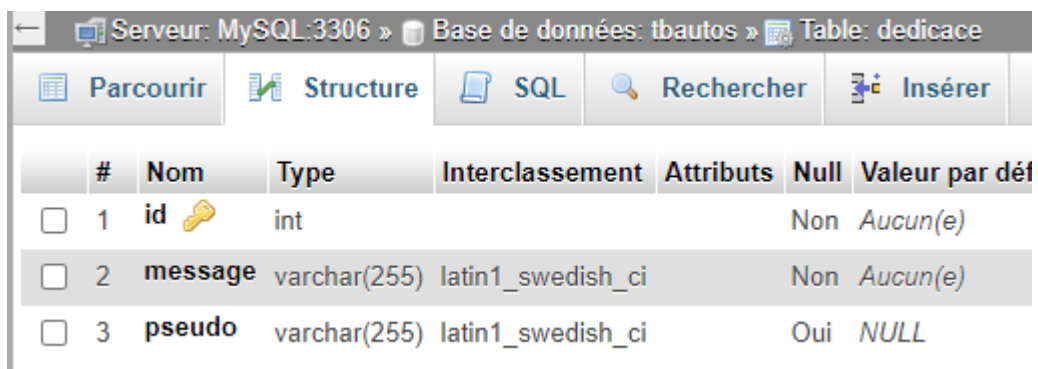
La class « . dedis » permet de créer le cadre :

```
.dedis {  
  height:50px;  
  width:100%;  
  background-color: #ffffff;  
  font-family:Arial;  
  font-size:16pt;  
  white-space: 200px;  
  color: #000000;  
  font-weight:bold;  
  border-style:solid;  
  border-color: #000000;  
}
```

J'utilise aussi les media queries afin que le cadre soit responsif en version mobile :

```
@media screen and (max-width: 740px) {  
  .dedis {  
    height:30px;  
    width:100%;  
    background-color: #ffffff;  
    font-family:Arial;  
    font-size:8pt;  
    white-space: 200px;  
    color: #000000;  
    font-weight:bold;  
    border-style:solid;  
    border-color: #000000;  
  }  
}
```

Par la suite, je crée la table dédicace dans la base de données :



#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut
1	id	int			Non	Aucun(e)
2	message	varchar(255)	latin1_swedish_ci		Non	Aucun(e)
3	pseudo	varchar(255)	latin1_swedish_ci		Oui	NULL

Message = Commentaire.

Pseudo = Nom Prénom, de la personne qui a publié le commentaire.



La fonction `dedicace()`; permet de gérer l'espace entre chaque avis et l'affichage des avis.

```
function dedicace(){
    $bdd=$bdd();
    // REQUEST
    $sql = 'SELECT * FROM `dedicace` ORDER BY id DESC LIMIT 10';

    // PREPARATION OF REQUEST
    $query = $bdd->prepare($sql);

    // EXECUTE OF REQUEST
    $query->execute();

    $reponse = $query->fetchAll(PDO::FETCH_ASSOC);

    foreach($reponse as $produit){
        echo " &nbsp; &nbsp; &nbsp; &nbsp; <font color='red'>".$produit['pseudo']."</font> : ".$produit['message']." ";
    }
}
```

Je récupère les informations de la table 'dedicace' à l'aide de la requête 'SELECT * FROM 'dedicace'' j'affiche les 10 derniers avis grâce à la requête, « ORDER BY id DESC LIMIT 10 » (sélectionne les 10 'id' en ordre décroissant).

Pour finir, la fonction `fetch()` ; permet d'afficher les informations de la table 'dedicace '

Je récupère les informations de la base de données dans le cadre :

```
 ".$produit['pseudo']."          ".$produit['message']." "
```

L'élément html '\$nbsp', permet de créer l'espace entre chaque avis

Résultat Frontend :



Résultat Frontend version mobile :





Une fois que j'ai terminé d'ajouter la barre de dédicace et les informations concernant le centre automobile, j'ai ajouté leurs numéros de téléphone avec une fonctionnalité pour l'utilisateur.

Lorsque l'utilisateur est sur version mobile et clique sur le numéro de téléphone, l'utilisateur pourra contacter directement l'entreprise :

```
<a class="b-phone__number" href="tel:+33766868480">+33 7 66 86 84 80</a>
```

VI – Contact :

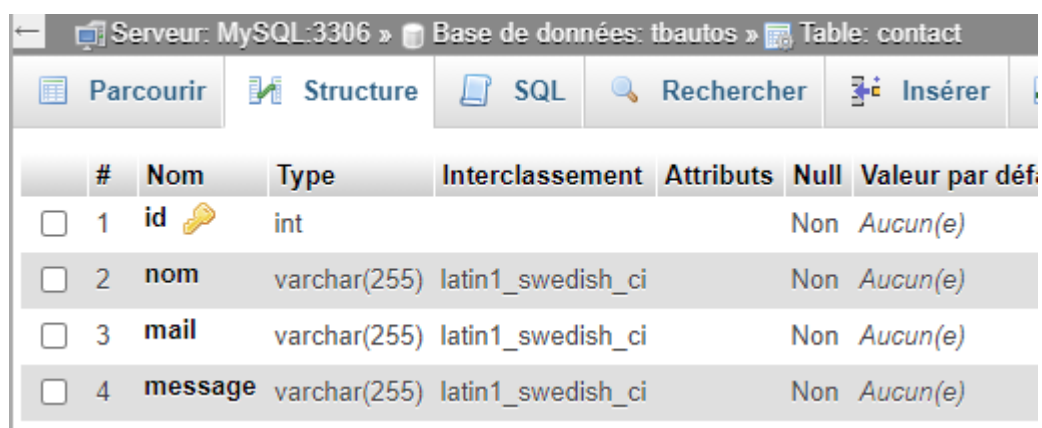
Pour la création de la page contact, j'ai intégré la localisation google maps du point de service de carte grise et un formulaire de contact.

Objectifs de la page contact : Lorsque le client à une question, il lui suffit de remplir le formulaire, mon maître de stage recevra les informations de la page contact dans l'administration que j'expliquerai par la suite.

Les éléments nécessaires afin de créer le formulaire de contact :

- Base de données
- Formulaire qui permettra d'écrire un avis
- Requête SQL qui permettra l'insertion dans la base de données

1 – Création de la table 'contact' dans la base de données :



The screenshot shows the MySQL Workbench interface with the 'contact' table structure displayed. The table has four columns: 'id' (int), 'nom' (varchar(255)), 'mail' (varchar(255)), and 'message' (varchar(255)). All columns are set to 'Non' for 'Null' and 'Aucun(e)' for 'Valeur par défaut'.

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut
1	id	int			Non	Aucun(e)
2	nom	varchar(255)	latin1_swedish_ci		Non	Aucun(e)
3	mail	varchar(255)	latin1_swedish_ci		Non	Aucun(e)
4	message	varchar(255)	latin1_swedish_ci		Non	Aucun(e)



nom = Nom Prénom du client

mail = Adresse mail du client

message = Message du client

2 – Création du formulaire de contact :

```
<form id="contactForm" method="post">
  <div class="form-group">
    <input class="form-control" type="text" name="nom" placeholder="Nom Prénom" required="required"/>
  </div>
  <div class="form-group">
    <input class="form-control" type="email" name="mail" placeholder="Adresse Mail"/>
  </div>
  <div class="form-group">
    <textarea class="form-control" rows="5" name="message" placeholder="Message" required="required"></textarea>
  </div>
  <button type="submit" value="Envoyer" name='send' class="btn btn-primary">Envoyer le message</button>
</form>
<?php addcontact(); ?>
```

Comme à la Page 12, je vérifie que le formulaire comporte les bonnes informations, afin que la fonction « addcontact() ; » puisse communiquer avec le formulaire qui utilise la méthode « POST ».

Nom Prénom, je lui donne le nom de 'nom'

Adresse Mail, je lui donne le nom de 'mail'

Message, je lui donne le nom de 'message'

Bouton 'ajouter', je lui donne le nom de 'send'.

Par la suite, j'appelle la fonction addcontact();

```
<?php addcontact(); ?>
```



Puis je vérifie la requête de la fonction addcontact() ; afin qu'il puisse communiquer avec le formulaire et que l'insertion du formulaire dans la base de données puisse fonctionner avec succès :

```
function addcontact(){
    $bdd=$bdd();
    if(isset($_POST['send'])&&!empty($_POST['nom'])&&!empty($_POST['mail'])&&!empty($_POST['message'])) {
        $req = $bdd->prepare('INSERT INTO contact (id, nom, mail, message) VALUES (NULL, :nom, :mail, :message)');
        $req->execute(array(
            'nom' => $_POST['nom'],
            'mail' => $_POST['mail'],
            'message' => $_POST['message'],));
        echo 'Votre message à bien était expédié';
    }
}
```

Les informations de ma requête SQL correspondent bien avec les informations du formulaire, je peux donc effectuer une insertion des informations à la base de données, une fois que j'appuie sur le bouton ajouter si tous se passe correctement, un message va s'afficher :

' Votre message à bien était expédié '.

III – Prise de rendez-vous :

Pour la création de la page rendez-vous, la communication entre la base de données et le formulaire est exactement comme le formulaire de contact c'est-à-dire une requête qui va effectuer l'insertion à la base de données.

1- Création de la table rendez-vous :

Serveur: MySQL:3306 » Base de données: tbautos » Table: rendezvous							
Parcourir Structure SQL Rechercher Insérer							
	#	Nom	Type	Interclassement	Attributs	Null	Valeur par
<input type="checkbox"/>	1	id	int			Non	Aucun(e)
<input type="checkbox"/>	2	nom	varchar(255)	latin1_swedish_ci		Non	Aucun(e)
<input type="checkbox"/>	3	telephone	varchar(255)	latin1_swedish_ci		Non	Aucun(e)
<input type="checkbox"/>	4	motif	varchar(255)	latin1_swedish_ci		Non	Aucun(e)
<input type="checkbox"/>	5	date	datetime			Non	Aucun(e)
<input type="checkbox"/>	6	suggestion	varchar(255)	latin1_swedish_ci		Non	Aucun(e)



Création du formulaire de prise de rendez-vous avec deux spécificités :

1- Un sélecteur qui permettra de choisir le motif du rendez-vous.

Carte grise

--Motif du rendez-vous--

Carte grise

Réparation ou entretien du véhicule

2 – Un calendrier qui permettra de choisir la date et l’heure du rendez-vous

Date et Heure du rendez vous:

jj/mm/aaaa --:--

avril 2021

lu ma me je ve sa di

29 30 31 1 2 3 4

5 6 7 8 9 10 11

12 13 14 15 16 17 18

19 20 21 22 23 24 25

26 27 28 29 30 1 2

3 4 5 6 7 8 9

Aujourd'hui

11 52

12 53

13 54

14 55

15 56

16 57

17 58

Informations concernant la création du formulaire en backend :

```
<form method="post">
  <div class="form-group">
    <input class="form-control" type="text" name="nom" placeholder="Nom Prénom" required="required"/>
  </div>
  <div class="form-group">
    <input class="form-control" type="text" name="telephone" placeholder="Numéro de téléphone" required="required"/>
  </div>
  <div class="form-group">
    <select class="form-control" name="motif">
      <option value="">--Motif du rendez-vous--</option>
      <option name="motif" value="Carte grise">Carte grise</option>
      <option name="motif" value="Réparation ou entretien du véhicule">Réparation ou entretien du véhicule</option>
    </select>
  </div>
  <div class="form-group">
    Date et Heure du rendez vous:
    <input class="form-control" type="datetime-local" name="date"/>
  </div>
  <div class="form-group">
    <textarea class="form-control" name="suggestion" rows="3" placeholder="Suggestion" required="required"></textarea>
  </div>
  <button name="sen" type="submit" class="btn btn-primary">Reserver</button>
</form>
<?php addcontactreparation(); ?>
```



Nom Prénom, je lui donne le nom de 'nom'

Téléphone, je lui donne le nom de 'telephone '

Motif du rendez-vous, je lui donne le nom de 'motif ', et deux valeurs lorsque l'utilisateur va sélectionner une des deux valeurs cela va s'envoyer à la base de données

Date et heure de rendez-vous, je lui donne le nom de 'date'

Suggestion, je lui donne le nom de 'suggestion'

Bouton 'Reserver', je lui donne le nom de 'sen'.

La balise '<select></select>' + '<option></option>' permet de créer un sélecteur

Le type 'datetime-local' permet de créer un calendrier.

Comme pour le formulaire de contact, je fais ensuite appelle à la fonction `addcontactreparation()` ; et je vérifie pour que l'insertion à la base de données puisse fonctionner :

```
function addcontactreparation(){
    $bdd=$bdd();
    if(isset($_POST['sen'])&&!empty($_POST['nom'])&&!empty($_POST['telephone'])&&!empty($_POST['motif'])&&!empty($_POST['date'])&&!empty($_POST['suggestion'])) {
        $req = $bdd->prepare('INSERT INTO rendezvous (id, nom, telephone, motif, date, suggestion) VALUES (NULL, :nom, :telephone, :motif, :date, :suggestion)');
        $req->execute(array(
            'nom' => $_POST['nom'],
            'telephone' => $_POST['telephone'],
            'motif' => $_POST['motif'],
            'date' => $_POST['date'],
            'suggestion' => $_POST['suggestion'],));

        echo 'Votre message a bien été envoyé';
        echo '<script language="Javascript">
        document.location.replace("rendezvous.php");
        </script>';
    }
}
```

Les noms correspondent bien avec les noms que j'ai mis dans le formulaire de prise de rendez-vous, l'insertion à la base de données peut donc fonctionner et le message peut s'afficher :

' Votre message à bien était expédié. '.

Dans un formulaire, il est nécessaire d'ajouter :

`required="required"`

Dans les inputs souhaités, pour que l'utilisateur soit obligé de remplir tous les champs importants.



Réalisation d'une administration

Pour la réalisation d'une administration, nous avons besoin de plusieurs éléments :

- Une Base de données
- Une template pour l'administration
- Une page de login pour sécuriser l'administration

Tout d'abord, j'ai créé une table dans la base de données 'admintb' :

The screenshot shows the MySQL table structure for 'admintb'. The table has 6 columns: id, pseudo, email, password, ip, and date_inscription. The 'id' column is the primary key and is of type int. The other columns are of type varchar or text, with 'date_inscription' being of type datetime. All columns are nullable and have a default value of 'Aucun(e)' or 'CURRENT_TIMESTAMP'.

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut
1	id	int			Non	Aucun(e)
2	pseudo	varchar(100)	latin1_swedish_ci		Non	Aucun(e)
3	email	varchar(100)	latin1_swedish_ci		Non	Aucun(e)
4	password	text	latin1_swedish_ci		Non	Aucun(e)
5	ip	varchar(20)	latin1_swedish_ci		Non	Aucun(e)
6	date_inscription	datetime			Non	CURRENT_TIMESTAMP

Ensuite le formulaire d'inscription :

```
<form action="inscription_traitement.php" method="post">
  <h2 class="text-center">Inscription</h2>
  <div class="form-group">
    <input type="text" name="pseudo" class="form-control" placeholder="Pseudo" required="required" autocomplete="off">
  </div>
  <div class="form-group">
    <input type="email" name="email" class="form-control" placeholder="Email" required="required" autocomplete="off">
  </div>
  <div class="form-group">
    <input type="password" name="password" class="form-control" placeholder="Mot de passe" required="required" autocomplete="off">
  </div>
  <div class="form-group">
    <input type="password" name="password_retype" class="form-control" placeholder="Re-tapez le mot de passe" required="required" autocomplete="off">
  </div>
  <div class="form-group">
    <button type="submit" class="btn btn-primary btn-block">Inscription</button>
  </div>
</form>
```

Pseudo, je lui donne le nom de 'pseudo'

Email, je lui donne le nom de 'email'

Mot de passe, je lui donne le nom de 'password'

Retapez le mot de passe, je lui donne le nom de 'password_retype'



Le formulaire d'inscription va fonctionner grâce à une requête qui se trouve dans le fichier inscription_traitement.php (Même principe qu'avec l'appelle d'une fonction).

inscription_traitement.php :

```
if(isset($_POST['pseudo']) && isset($_POST['email']) && isset($_POST['password']) && isset($_POST['password_retype']))
{
    $pseudo = htmlspecialchars($_POST['pseudo']);
    $email = htmlspecialchars($_POST['email']);
    $password = htmlspecialchars($_POST['password']);
    $password_retype = htmlspecialchars($_POST['password_retype']);

    $check = $bdd->prepare('SELECT pseudo, email, password FROM utilisateurs WHERE email = ?');
    $check->execute(array($email));
    $data = $check->fetch();
    $row = $check->rowCount();

    if($row == 0){
        if(strlen($pseudo) <= 100){
            if(strlen($email) <= 100){
                if(filter_var($email, FILTER_VALIDATE_EMAIL)){
                    if($password == $password_retype){

                        $password = hash('sha256', $password);
                        $ip = $_SERVER['REMOTE_ADDR'];

                        $insert = $bdd->prepare('INSERT INTO utilisateurs(pseudo, email, password, ip) VALUES(:pseudo, :email, :password, :ip)');
                        $insert->execute(array(
                            'pseudo' => $pseudo,
                            'email' => $email,
                            'password' => $password,
                            'ip' => $ip
                        ));
                    }
                }
            }
        }
    }
}
```

Comme pour les fonctions que j'ai expliquées en haut, je vérifie les noms afin que la requête puisse communiquer avec le formulaire d'inscription.

J'utilise une méthode de hachage de mot de passe.

Le mot de passe est crypté en SHA256

Le chiffrement **SHA256**, comme toute fonction de **hachage**, étant basé sur des fonctions non-linéaires (non réversibles), il n'existe pas de méthode de déchiffrement.

J'enregistre aussi la date de l'inscription et l'adresse IP de la personne qui s'est inscrit.

On enregistre l'adresse IP grâce \$_SERVER['REMOTE_ADDR'], et la date et l'heure de l'inscription grâce au type date-time dans la base de données.



J'ai créé un formulaire de connexion qui va récupérer les informations de l'inscription dans la base de données et vérifier si les informations sont correctes pour permettre de créer une session et d'accéder à l'administration :

```
<form action="connexion.php" method="post">
  <h1 id="litheader">TBAUTOS Administration</h1>
  <div class="inset">
    <p>
      <input type="text" name="email" id="email" placeholder="Adresse Email">
    </p>
    <p>
      <input type="password" name="password" id="password" placeholder="Mot de Passe">
    </p>
    <input class="loginLoginValue" type="hidden" name="service" value="login" />
  </div>
  <p class="p-container">
    <input type="submit" name="Login" id="go" value="Connexion">
  </p>
</form>
```

Comme pour l'inscription, le formulaire de connexion va communiquer avec une requête qui se trouve dans connexion.php et permettre de créer une session sécurisée pour accéder à l'administration.

```
session_start();
require_once 'config.php';

if(isset($_POST['email']) && isset($_POST['password']))
{
    $email = htmlspecialchars($_POST['email']);
    $password = htmlspecialchars($_POST['password']);

    $check = $bdd->prepare('SELECT pseudo, email, password FROM adminnb WHERE email = ?');
    $check->execute(array($email));
    $data = $check->fetch();
    $row = $check->rowCount();

    if($row == 1)
    {
        if(filter_var($email, FILTER_VALIDATE_EMAIL))
        {
            $password = hash('sha256', $password);
            if($data['password'] === $password)
            {
                $_SESSION['user'] = $data['email'];
                header('Location: ../administration.php');
                die();
            }
        }
    }
}
```

Lorsque je vais rentrer mes informations dans le formulaire de connexion, la requête va sélectionner le pseudo, l'adresse mail et le mot de passe dans la table 'adminnb', selon l'adresse mail renseignée d'où la syntaxe 'WHERE email = ?'.



Si les informations renseignées sont correctes, la session va être créée et je serais redirigé vers le fichier administration.php.

Il faut sécuriser le fichier administration.php, car il peut y avoir une personne lambda qui va essayer d'accéder au fichier administration.php sans passer par le formulaire de connexion.

Pour cela, je rédige la requête suivant au début de chaque fichier que je souhaite protéger :

```
session_start();
if(!isset($_SESSION['user']))
    header('Location:login/');
```

La page va vérifier la session de connexion, si une personne lambda essaie d'accéder sans passer par le formulaire de connexion alors il sera redirigé automatiquement vers le formulaire de connexion.

Lors de la connexion, si l'administrateur se trompe de mot de passe ou d'adresse mail, j'affiche un message d'erreur :

```
}else header('Location: index.php?login_err=password');
}else header('Location: index.php?login_err=email');
}else header('Location: index.php?login_err=already');
```

```
if(isset($_GET['login_err']))
{
    $err = htmlspecialchars($_GET['login_err']);
    switch($err)
    {
        case 'password':
            <?php
            <div class="alert alert-danger">
                <strong>Erreur</strong> mot de passe incorrect
            </div>
            <?php
            break;

        case 'email':
            <?php
            <div class="alert alert-danger">
                <strong>Erreur</strong> email incorrect
            </div>
            <?php
            break;

        case 'already':
            <?php
            <div class="alert alert-danger">
                <strong>Erreur</strong> compte non existant
            </div>
            <?php
            break;
```



Cela permettra de donner des indices à l'administrateur et de lui faciliter la connexion à l'administration.

Une fois que mon maître de stage s'est inscrit j'ai enlevé les fichiers inscription.php et inscription_traitement.php du ftp pour qu'aucune personne ne puisse s'inscrire et accéder à l'administration à part mon maître de stage.

Les instructions que je devais suivre pour l'administration :

- Ajouter un véhicule
- Modifier ou supprimer un véhicule
- Modifier ou supprimer les avis à la page d'accueil
- Modifier ou supprimer les avis à la page de réparation
- Boîte de réception (Rubrique contact du site vitrine)
- Voir les prises de rendez-vous (Rubrique prise de rendez-vous)

Ajouter d'un véhicule :

Pour l'ajout d'un véhicule, l'administrateur aura besoin d'ajouter :

Toutes les informations sur le véhicule et les 5 images (Demandé par mon maître de stage).

Pour cela, je dois créer un formulaire d'ajout et une requête SQL (se trouvant dans le fichier upload.php) qui permettra d'ajouter les 5 images dans un dossier situé dans le ftp et les autres informations sur le véhicule.

Formulaire d'ajout partie 1 :

```
<form name="ajout" action="upload.php" method="POST" enctype="multipart/form-data">
  <div class="form-group">
    <label class="col-md-12">Selectionner l'image 1 à télécharger :</label>
    <div class="col-md-12">
      <input type="file" name="image_voitures" id="fileToUpload" ><br>
    </div><br>
    <div class="form-group">
      <label class="col-md-12">Selectionner l'image 2 à télécharger :</label>
      <div class="col-md-12">
        <input type="file" name="image_two" id="fileToUpload" ><br>
      </div><br>
    <div class="form-group">
      <label class="col-md-12">Selectionner l'image 3 à télécharger :</label>
      <div class="col-md-12">
        <input type="file" name="image_three" id="fileToUpload" ><br>
      </div><br>
    <div class="form-group">
      <label class="col-md-12">Selectionner l'image 4 à télécharger :</label>
      <div class="col-md-12">
        <input type="file" name="image_four" id="fileToUpload" ><br>
      </div><br>
    <div class="form-group">
      <label class="col-md-12">Selectionner l'image 5 à télécharger :</label>
      <div class="col-md-12">
        <input type="file" name="image_five" id="fileToUpload" ><br>
      </div><br>
  </div></div>
```



Partie 2 :

```
<div class="form-group">
<label class="col-md-12">Titre de l'annonce</label>
<div class="col-md-12">
<input type="text" id="nom_voitures" name="nom" class="form-control form-control-line">
</div><br>
<div class="form-group">
<label class="col-md-12">Prix du véhicule</label>
<div class="col-md-12">
<input type="text" id="prix_voitures" name="prix" class="form-control form-control-line">
</div><br>
<div class="form-group">
<label class="col-md-12">Nombre de kilomètres</label>
<div class="col-md-12">
<input type="text" id="km_voitures" name="km" class="form-control form-control-line">
</div><br>
<label class="col-md-12">Carburant</label>
<div class="col-md-12">
<input type="text" id="carburant_voitures" name="carburant" class="form-control form-control-line">
</div><br>
<label class="col-md-12">Année du véhicule</label>
<div class="col-md-12">
<input type="text" id="annee_voitures" name="annee" class="form-control form-control-line">
</div><br>
<label class="col-md-12">Boîte à vitesse</label>
<div class="col-md-12">
<input type="text" id="boite_voitures" name="boite" class="form-control form-control-line">
</div><br>
<label class="col-md-12">Description du véhicule</label>
<div class="col-md-12">
<textarea type="text" id="description_voitures" name="description" class="form-control form-control-line"></textarea>
</div><br>
<div class="col-md-12"><br>
<button class="btn btn-warning" type="submit" name="ajouter" value="ok">Ajouter</button>
</form>
```

Le formulaire d'ajout, communique avec 5 requêtes disponibles dans le page upload.php,

Ses 5 requêtes vont permettre d'ajouter les 5 images dans un dossier que j'ai créé dans le ftp, une fois les 5 requêtes terminées, la fonction ajouterv() ; va permettre d'ajouter le chemin de chaque images dans la base de données, afin de permettre l'affichage de celle-ci sur le site vitrine.

Exemple d'une requête :

```
$target_dir = "voiture/";
$target_file = $target_dir . basename($_FILES["image_five"]["name"]);
$uploadOk = 1;
$imageFileType = strtolower(pathinfo($target_file,PATHINFO_EXTENSION));

// Check if $uploadOk is set to 0 by an error
if ($uploadOk == 0) {
    echo "Désolé, votre fichier n'a pas été téléchargé.";
    // if everything is ok, try to upload file
} else {
    if (move_uploaded_file($_FILES["image_five"]["tmp_name"], $target_file)) {
        $nomimg5 = $target_dir . "htmlspecialchars( basename( $_FILES["image_five"]["name"]));
        echo "Le fichier ". htmlspecialchars( basename( $_FILES["image_five"]["name"])). " a été téléchargé.";
        $delai=1;
        $url='administration.php';
        header("Refresh: $delai;url=$url");
    } else {
        echo "Désolé, une erreur s'est produite lors du téléchargement de votre fichier.";
    }
}
ajouterv($nomimg, $nomimg2, $nomimg3, $nomimg4, $nomimg5);
```



L'image va être stockée dans le dossier 'voiture'.

Ensuite la fonction `ajouterv()` va intervenir pour la communication avec la base de données :

```
function ajouterv($ajoutimg, $ajoutimg2, $ajoutimg3, $ajoutimg4, $ajoutimg5){
    $bdd=bdd();
    if(isset($_POST['ajouter'])&&!empty($_POST['nom'])&&!empty($_POST['prix'])&&!empty($_POST['km'])&&!empty($_POST['carburant'])
    $req = $bdd->prepare('INSERT INTO voitures(id_voitures, image_voitures, image_two, image_three, image_four, image_five, nom, prix, km, carburant, annee, boite, description) VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?)');
    $req->execute(array(
        'image' => $ajoutimg,
        'image2' => $ajoutimg2,
        'image3' => $ajoutimg3,
        'image4' => $ajoutimg4,
        'image5' => $ajoutimg5,
        'nom' => $_POST['nom'],
        'prix' => $_POST['prix'],
        'km' => $_POST['km'],
        'carburant' => $_POST['carburant'],
        'annee' => $_POST['annee'],
        'boite' => $_POST['boite'],
        'description' => $_POST['description'],));
    echo 'Le nouveau véhicule a été ajouté !';
}
}
```

Une fois que tout est ajouté à la base de données, je pourrais afficher des véhicules sur le système que j'ai créé précédemment à la page 15.

Mon maître de stage m'a demandé de créer un système qui lui permettra de modifier ou de supprimer un véhicule.

Pour cela, je vais créer une fonction pour afficher les informations d'un véhicule :

```
function updatevehicule(){
    $bdd=bdd();
    $recuperation = $bdd->prepare('SELECT * FROM voitures');
    $recuperation->execute();
    while ($stache = $recuperation->fetch()) {
        echo "<form>";
        <div>
        <center>
        Id : <input type='text' class='form-control' name='id_voitures' value='".$stache['id_voitures']."'><br>
        <center><img class='adminv' src='".$stache['image_voitures']."'></center><br><br>
        Titre de l'annonce : <input type='text' class='form-control' name='nom_voitures' value='".$stache['nom_voitures']."'>
        Prix du véhicule : <input type='text' class='form-control' size='20' name='prix_voitures' value='".$stache['prix_voitures']."'>
        Carburant : <input type='text' class='form-control' size='20' name='carburant_voitures' value='".$stache['carburant_voitures']."'>
        Année : <input type='text' class='form-control' size='20' name='annee_voitures' value='".$stache['annee_voitures']."'>
        Boite : <input type='text' class='form-control' size='20' name='boite_voitures' value='".$stache['boite_voitures']."'>
        Description: <input type='text' class='form-control' size='20' name='description_voitures' value='".$stache['description_voitures']."'>

        <button type='submit' class='btn btn-success mt-1' value='modifier' name='modifier'>Modifier</button>
        <button type='submit' class='btn btn-danger mt-1' value='supprimer' name='delete'>Supprimer</button></center><br /><br><br>

        </form>
        </div>";
    }
}
```

Une fois que j'ai récupéré les informations grâce à la fonction `fetch()` ;



Je vais donner un nom à chaque input et créer une requête qui permettra de modifier les informations qu'il y a dans chaque input :

```
if(isset($_GET['modif'])&&empty($_GET['id_voitures'])&&empty($_GET['nom_voitures'])&&empty($_GET['prix_voitures'])&&empty($_GET['carburant_voitures'])&&empty($_GET['annee_voitures'])&&empty($_GET['boite_voitures'])&&empty($_GET['description_voitures']))  
{  
    $id=$_GET['id_voitures'];  
    $req = $bdd->prepare('UPDATE voitures SET nom_voitures = :nom_voitures, prix_voitures = :prix_voitures, carburant_voitures = :carburant_voitures, annee_voitures = :annee_voitures, boite_voitures = :boite_voitures, description_voitures = :description_voitures WHERE id_voitures = :id_voitures');  
    $req->execute(array(  
        'nom_voitures' => $_GET['nom_voitures'],  
        'prix_voitures' => $_GET['prix_voitures'],  
        'carburant_voitures' => $_GET['carburant_voitures'],  
        'annee_voitures' => $_GET['annee_voitures'],  
        'boite_voitures' => $_GET['boite_voitures'],  
        'description_voitures' => $_GET['description_voitures'],  
        ':id_voitures' => $id ));  
    echo "la modification a été effectuée ";  
    echo '<script language="Javascript">  
    document.location.replace("adminsuppr.php");  
</script>';  
}
```

Grâce au nom que j'ai donné dans chaque input précédemment, la modification pourra être effectuée avec succès.

Pour finir, je vais créer une fonction delete() ; pour supprimer un véhicule :

```
function deletevehicule(){  
    $bdd=$bdd();  
    if(isset($_GET['delete'])&&empty($_GET['id_voitures']))  
    {  
        $id=$_GET['id_voitures'];  
        $req = $bdd->prepare('DELETE FROM voitures WHERE id_voitures= :id_voitures');  
        $req->execute(array(  
            ':id_voitures' => $id ));  
        echo "l'intervention a été supprimé.";  
        echo '<script language="Javascript">  
        document.location.replace("adminsuppr.php");  
        </script>';  
    }  
}
```

Pour supprimer un véhicule, la requête va communiquer avec le bouton que j'ai nommé 'delete', et va supprimer l'id du véhicule. Étant donné que l'id est équivalent à toutes les informations d'un véhicule, le véhicule et tous ses informations seront supprimées de la base de données.



Résultat en Frontend :

Id : 56



Titre de l'annonce : Bmw 650ci

Prix du véhicule : 19 500

Carburant : Essence

Année : 03/2007

Boite : Automatique

Description : BMW Série 6 4.8 367 CH 650 CI 105350 km, FINITION

[Modifier](#) [Supprimer](#)

Pour le système ajouter ou supprimer un avis cela reste exactement le même système.

Pour la boîte de réception ou les prises de rendez-vous dans l'administration, cela reste aussi la même requête, je vais prendre l'exemple de la page contact.

L'utilisateur a une question concernant un véhicule, il va donc se rendre à la page contact.

Une fois que le formulaire est rempli, la requête va s'exécuter et les informations seront envoyées dans la table contact.

Dans l'administration, je vais récupérer les informations de la table 'contacts' et les affichés avec la fonction fetch() ; :

```
function contact(){
    $bdd=$bdd();
    $recuperation = $bdd->prepare('SELECT * FROM contact');
    $recuperation->execute();
    while ($tache = $recuperation->fetch()) {
        echo '
        <div><h4>id : '.$tache['id'].'</h4>
        <h4>Nom Prénom : '.$tache['nom'].'</h4>
        <h4>Adresse Mail : <a href="mailto:'.$tache['mail'].'"> '.$tache['mail'].'</h4> </a>
        <h4>Message : '.$tache['message'].'</h4></div>

        <hr>

        '
    }
}
```



Présentation du jeu d'essai de la fonctionnalité la plus représentative






Jeu d'essai de la fonctionnalité :

Je test à présent le formulaire d'ajout d'un véhicule comme je l'ai fait précédemment à la page 29.

Tout d'abord, j'ai 5 images en .jpg dans un dossier de mon bureau, l'objectif de vérifier que lorsque j'ajoute les 5 images dans le formulaire d'ajout, cela s'ajoutera dans le dossier « voiture/ » situé dans mon ftp, et le chemin qui nous permettra d'afficher les images, il faut que le chemin (url où se trouve l'image), s'affiche automatiquement dans la base de données.

Test du formulaire d'ajout :

Les 5 images en .jpg sont situées dans le fichier de mon bureau d'ordinateur :

 access1	30/04/2021 12:23	Fichier JPG
 access2	30/04/2021 12:23	Fichier JPG
 access3	30/04/2021 12:23	Fichier JPG
 access4	30/04/2021 12:23	Fichier JPG
 access5	30/04/2021 12:24	Fichier JPG

J'ajoute les 5 images et je complète le formulaire :

Selectionner l'image 1 à télécharger :

Choisir un fichier access1.jpg

Selectionner l'image 2 à télécharger :

Choisir un fichier access2.jpg

Selectionner l'image 3 à télécharger :

Choisir un fichier access3.jpg

Selectionner l'image 4 à télécharger :

Choisir un fichier access4.jpg

Selectionner l'image 5 à télécharger :

Choisir un fichier access5.jpg



Si ce message s'affiche, c'est que tout a été complété avec succès.

Le fichier access1.jpg a été téléchargé. Le fichier access2.jpg a été téléchargé. Le fichier access3.jpg a été téléchargé. Le fichier access4.jpg a été téléchargé. Le fichier access5.jpg a été téléchargé.

Je vérifie ensuite si les images ont bien été ajoutées dans le dossier « voiture / » du ftp :

C > Disque local (C:) > wamp64 > www > tb > voiture



Pour finir, je vérifie que le chemin a bien été ajouté dans la base de données :

image_voitures image_two image_three image_four image_five

voiture/access1.jpg voiture/access2.jpg voiture/access3.jpg voiture/access4.jpg voiture/access5.jpg

Le test fonctionne.



Veille effectuée durant le projet sur les vulnérabilités sécurité

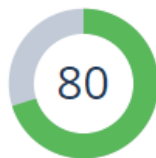
Lorsque j'ai terminé le site, j'ai procédé à l'hébergement chez 'allsheberg', une fois que le site est en ligne, je me suis rendu compte que le lien était : <http://tbauto.fr> et non sécurisé.

J'ai contacté l'hébergeur qui m'a envoyé un tutoriel pour installer un certificat SSL, et une fois le certificat installé, j'ai configuré le fichier .htaccess pour que le site soit redirigé automatiquement vers : <https://tbautos.fr>

Pour tester la sécurité du site, je me suis rendu sur <https://www.ionos.fr/> :

Résultats pour tbautos.fr

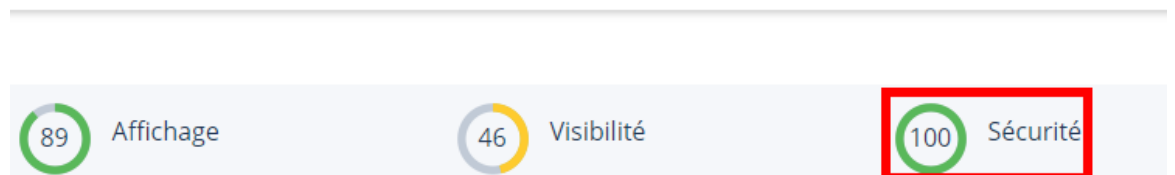
[Recommencer l'analyse](#) ou [Vérifier un autre site Web](#)



Résultat global

Votre site peut être amélioré.

Suivez les conseils de l'outil d'analyse de site, afin d'optimiser votre présence et votre visibilité.





S'il y a un problème d'attaque ou de sécurité, l'administrateur prévient directement depuis le site comme le souci du 30/04 :

Sébastien D. a commenté le 30.04.2021 12:36	Admin
Une attaque est en cours, le flux est en cours de mitigation, au vu de l'importance de l'attaque des flux passent a travers, des ajustements sont en cours pour fluidifier le trafic.	
Sébastien D. a commenté le 30.04.2021 12:48	Admin
Seul les adresses IP en 154.44.177.0/24 sont actuellement impacté, les machines tournant sur le bloc 193.26.21.0/24 continue de fonctionner actuellement.	
Sébastien D. a commenté le 30.04.2021 12:49	Admin
L'ensemble des machines remonte progressivement.	
Sébastien D. a commenté le 30.04.2021 12:51	Admin
<p>LeJack5 est de nouveau UP. PSK1 également.</p> <p>Si votre VPS rencontre des problèmes de connexions, contacter notre support.</p> <p>Nous maintenons une surveillance active.</p>	
Sébastien D. a commenté le 30.04.2021 12:54	Admin
Nous avons de nouveau des pertes de paquet, le filtrage est en cours	

J'utilise aussi le site <https://www.cert.ssi.gouv.fr/> pour être informé :

Sur le site **CERT-FR**, Centre gouvernemental de veille, d'alerte et de réponse aux attaques informatiques, ils communiquent sur les rapports de Menaces et Incidents de **l'ANSSI**, les alertes de sécurité et les avis de sécurité qui font état des vulnérabilités trouvées et comment s'en prémunir.



Situation de travail ayant nécessité une recherche à partir de site anglophone

Pour le live chat, j'ai choisi l'application tawk.to

Tawk.to est une application gratuite qui vous permet de dialoguer en temps réel (en live chat) avec les visiteurs de votre site. Il ne faut que quelques minutes pour installer cette application directement sur votre site soit en copiant une simple ligne de code, pour l'installation du widget le site est en anglais :

Monitor customer activity in real time

See which pages your visitors are on in real time, what they are searching for within your Knowledge Base and how often they visit your website. View activity across their entire journey. See clearly which team member responded to a customer last and maintain a complete conversation history.

for the installation of a widget, you must place this line of code **under your header**

```
<!--Start of Tawk.to Script-->
<script type="text/javascript">
var Tawk_API=Tawk_API||{}, Tawk_LoadStart=new Date();
(function(){
var
s1=document.createElement("script"),s0=document.getElementsByTagName("script")[0];
s1.async=true;
s1.src='https://embed.tawk.to/603e54ab1c1c2a130d640d83/1evpnlfo1';
s1.charset='UTF-8';
s1.setAttribute('crossorigin','*');
s0.parentNode.insertBefore(s1,s0);
})();
</script>
<!--End of Tawk.to Script-->
```



Traduction :

Surveillez l'activité des clients en temps **réel**

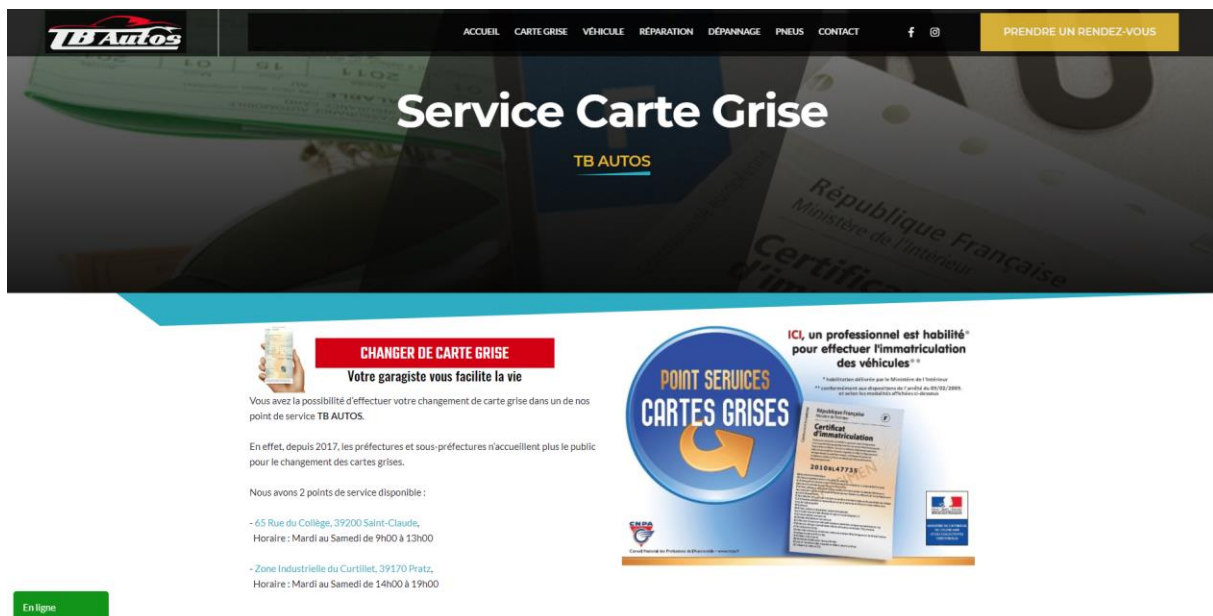
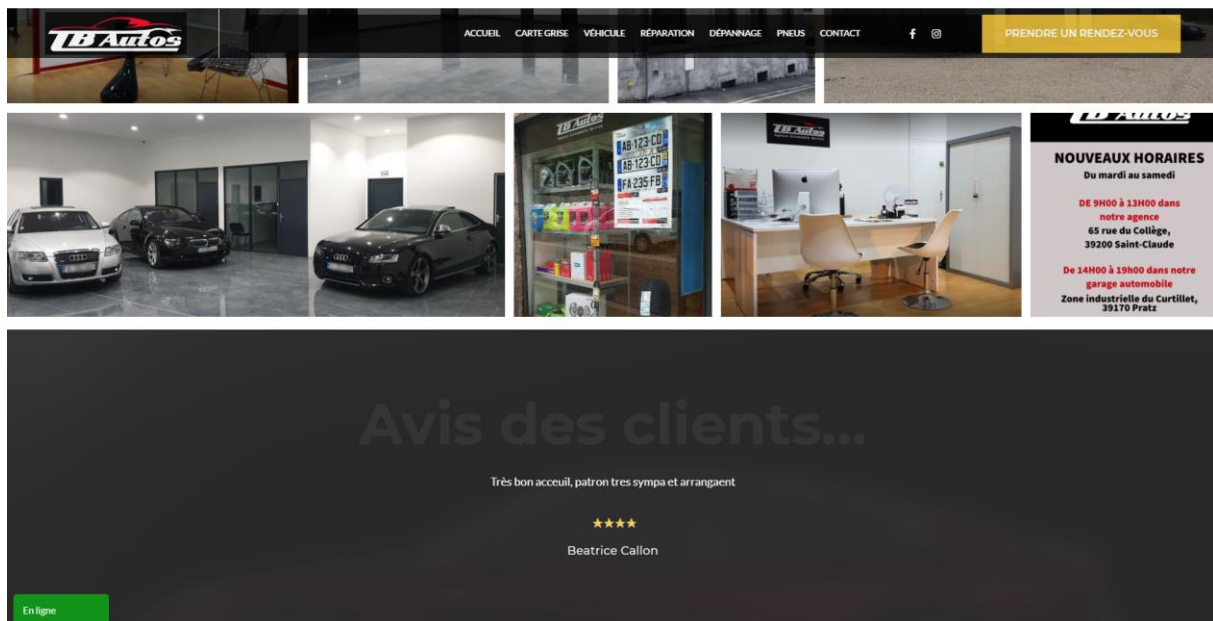
Découvrez les pages sur lesquelles vos visiteurs se trouvent en temps réel, ce qu'ils recherchent dans votre base de connaissances et la fréquence à laquelle ils visitent votre site Web. Affichez l'activité tout au long de leur parcours. Voyez clairement quel membre de l'équipe a répondu à un client en dernier et maintenez un historique complet des conversations.

Pour l'installation d'un widget, vous devez placer cette ligne de code **sous votre header**

```
<!--Start of Tawk.to Script-->
<script type="text/javascript">
var Tawk_API=Tawk_API||{}, Tawk_LoadStart=new Date();
(function(){
var
s1=document.createElement("script"),s0=document.getElementsByTagName("scri
pt")[0];
s1.async=true;
s1.src='https://embed.tawk.to/603e54ab1c1c2a130d640d83/1evpnlfo1';
s1.charset='UTF-8';
s1.setAttribute('crossorigin','*');
s0.parentNode.insertBefore(s1,s0);
})();
</script>
<!--End of Tawk.to Script-->
```



Annexes





Bienvenue au Centre Automobile TBAUTOS

Le garage effectue les réparations et l'entretien de votre véhicule. TBAUTOS offre un service de qualité pour tous les types de véhicules. Tous les services mécaniques sont effectués par des mécaniciens qualifiés et nous offrons une gamme complète de services de garage pour répondre à toutes les demandes de notre clientèle.

+33 7 66 86 84 80

En ligne

