
TITRE PROFESSIONNEL DÉVELOPPEUR WEB ET WEB MOBILE

EL HASNAOUI ZAHRA

2023

REMERCIEMENT

Je tiens sincèrement à exprimer ma profonde gratitude envers SJT pour m'avoir ouvert les portes de cette précieuse formation de développeur Web et Web mobile.

Je tiens à adresser des remerciements tout spéciaux à ma formatrice, Mme BAH Fatoumata, dont l'enseignement, la disponibilité, et la bienveillance inestimable m'ont accompagné tout au long de ce parcours. Sa présence et son dévouement ont été des piliers essentiels dans ma réussite.

Je n'oublie pas de souligner l'importante contribution de la région des Hauts-de-France, qui a rendu cette formation possible grâce à son financement.



SOMMAIRE

| | |
|--|----|
| Remerciement | 2 |
| Liste des compétences du référentiel | 4 |
| Résumé du projet | 5 |
| Résumé du projet en anglais | 6 |
| Cahier des charges, expression des besoins, ou spécifications fonctionnelles du projet | 7 |
| Spécifications techniques du projet, élaborées par le candidat, y compris pour la sécurité et le web mobile | 10 |
| Développer la partie Front-end d'une application web ou web mobile en intégrant les recommandations de sécurité. | |
| Maquetter une application | 10 |
| Réalisations du candidat comportant les extraits de code les plus significatifs et en les argumentant, y compris pour la sécurité et le web mobile | 12 |
| Réaliser une interface utilisateur web statique et adaptable | 12 |
| SEO | 16 |
| Développer une interface utilisateur web dynamique | 17 |
| Développer la partie Back-end d'une application web ou web mobile en intégrant les recommandations de sécurité. | |
| Créer une base de données | 21 |
| Architecture et Structure | 29 |
| Développer les composants d'accès aux données | 30 |
| Développer la partie Back-end d'une application web ou web | 34 |
| Présentation du jeu d'essai élaboré par le candidat de la fonctionnalité la plus représentative | 37 |
| Description de la veille, effectuée par le candidat durant le projet, sur les vulnérabilités de sécurité | 62 |
| Description d'une situation de travail ayant nécessité une recherche, effectuée par le candidat durant le projet, à partir de site anglophone | 63 |
| Lexique | 68 |
| Conclusion | 70 |

Liste des compétences du référentiel qui sont couvertes par le projet

Développer la partie Front-end d'une application web ou web mobile en intégrant les recommandations de sécurité.

1. Maquetter une application :

Maquettage de mon projet sur le logiciel **Figma**.

2. Réaliser une interface utilisateur web statique et adaptable :

Utilisation des langages **HTML** et **CSS** pour réaliser la mise en page de mon projet.

3. Développer une interface utilisateur web dynamique

Utilisation du langage **Javascript** pour dynamiser les pages.

Développer la partie Back-end d'une application web ou web mobile en intégrant les recommandations de sécurité.

1. Créer une base de données :

Utilisation du logiciel **XAMPP** pour faciliter la création de ma base de données et des différents liens entre les tables puis passage sur **MySQL** pour la mise en place.

2. Développer les composants d'accès aux données :

Utilisation du **PHP Data Objects** (PDO) et du langage PHP pour mettre en relation les pages de mon site et **MySQL**.

3. Développer la partie Back-end d'une application web ou web mobile :

Exécuter des requêtes **SQL** pour interagir entre **PHP** et **MySQL**.

Résumé du projet

Mon projet est un site de réservation de taxi, suite à plusieurs années d'expérience en tant que conductrice de taxi et taxi conventionné, j'ai été profondément immergé dans les réalités de ce secteur. Transportant fréquemment des personnes malades nécessitant des trajets médicaux réguliers, j'ai développé une compréhension aiguë des besoins spécifiques de ce public. Cette expérience m'a inspiré à créer un site de réservation de taxi qui non seulement améliore la vie des passagers, mais simplifie également la gestion quotidienne des chauffeurs.

Conscient du fardeau administratif imposé aux taxis conventionnés pour s'assurer de la validation des trajets médicaux, j'ai mis en place sur mon site une disposition clé : l'obligation pour les passagers de fournir un bon de transport médical valide lors de la réservation. Cette démarche vise à alléger le processus pour les chauffeurs qui, dans le passé, devaient parfois courir après les clients ou leurs collègues pour obtenir ce document vital. En automatisant cette exigence, je vise à simplifier la vie des chauffeurs, tout en garantissant aux passagers une expérience de réservation fluide et conforme aux exigences médicales.

En parallèle, j'ai introduit un système d'empreinte de carte bancaire pour les réservations, tant professionnelles que personnelles. Cette mesure a un double avantage. D'un côté, elle protège les chauffeurs en évitant les pertes financières causées par des annulations de dernière minute non signalées. D'un autre côté, elle offre aux passagers une solution pratique qui élimine la nécessité de fournir des informations de paiement à chaque réservation. Cette fonctionnalité est conçue pour renforcer la confiance et l'engagement des passagers tout en optimisant la planification des trajets pour les chauffeurs.

Afin de parachever cette initiative centrée sur l'utilisateur, j'ai mis en place un espace client personnalisé. Grâce à cette fonctionnalité, les utilisateurs peuvent créer un compte sur le site et accéder à leurs réservations passées et futures en un seul endroit. Cette centralisation des informations contribue à une expérience de réservation plus fluide et organisée pour les passagers.

En conclusion, ma décision de créer ce site de réservation de taxi s'appuie sur ma riche expérience en tant que chauffeur conventionné, avec une attention particulière aux transports médicaux. Mon objectif est double : améliorer la vie des passagers en simplifiant leurs démarches administratives et faciliter le quotidien des chauffeurs en automatisant les processus clés. En tant que développeur web, je m'engage à mettre en œuvre une plateforme qui répond efficacement aux besoins des utilisateurs, tout en optimisant l'ensemble du processus de réservation et de transport.

Summary project

My project is a taxi booking site, following several years of experience as a taxi driver and contracted taxi, I have been deeply immersed in the realities of this sector. Frequently transporting sick people requiring regular medical trips, I developed an acute understanding of the specific needs of this public. This experience inspired me to create a taxi booking site that not only improves passengers' lives, but also simplifies the daily management of drivers.

Aware of the administrative burden imposed on approved taxis to ensure the validation of medical journeys, I have implemented a key provision on my site: the obligation for passengers to provide a valid medical transport voucher when booking. . This move aims to ease the process for drivers who, in the past, sometimes had to chase after customers or co-workers to obtain this vital document. By automating this requirement, I aim to simplify the lives of drivers, while guaranteeing passengers a smooth booking experience that complies with medical requirements.

At the same time, I introduced a credit card imprint system for reservations, both professional and personal. This measure has a double advantage. On the one hand, it protects drivers by avoiding financial losses caused by unreported last-minute cancellations. On the other hand, it offers passengers a convenient solution that eliminates the need to provide payment information with each reservation. This feature is designed to build passenger confidence and engagement while optimizing trip planning for drivers.

To complete this user-centric initiative, I set up a personalized customer area. With this feature, users can create an account on the site and access their past and future bookings in one place. This centralization of information contributes to a smoother and more organized booking experience for passengers.

In conclusion, my decision to create this taxi reservation site is based on my rich experience as a licensed driver, with particular attention to medical transport. My objective is twofold: to improve the lives of passengers by simplifying their administrative procedures and to facilitate the daily lives of drivers by automating key processes. As a web developer, I am committed to implementing a platform that effectively meets user needs, while optimizing the entire booking and transport process.

Cahier des charges, expression des besoins, ou spécifications fonctionnelles du projet

Cahier des charges

Introduction

La création de cette plateforme Zine Taxi aura pour but de faciliter la réservation d'un taxi pour tout type de transport : professionnel, médical ou personnel.

Le site sera aussi bien accessible depuis un navigateur web sur ordinateur ou sur mobile.

Description de la demande

Le site doit pouvoir être utilisé facilement par des utilisateurs de tous âges afin de faciliter la réservation. De plus, l'objectif est de garantir que chaque réservation soit complète, incluant tous les documents nécessaires, afin de simplifier la gestion administrative de nos chauffeurs.

La barre de navigation située en haut du site permettra d'un simple clic d'accéder aux différentes fonctionnalités.

Le site est composé de différentes pages clés :

- Page Accueil

La page d'accueil offre une présentation brève de nos diverses options de transport, allant du transport professionnel au transport médical et personnel. Elle est enrichie de trois animations photos distinctes pour chaque type de transport, chacune offrant un accès direct à la réservation lorsqu'elle est cliquée. De plus, nous avons intégré une fonctionnalité d'API météo, permettant aux utilisateurs de consulter les prévisions météorologiques par ville avant de finaliser leur réservation, garantissant ainsi un voyage en toute sérénité.

- Page Réservation

Les utilisateurs doivent s'inscrire ou se connecter avant de pouvoir effectuer une réservation.

Une fois connectés, ils peuvent sélectionner le type de transport, saisir les détails de leur trajet (lieu de départ, destination, date et heure...).

Les réservations confirmées sont ajoutées à l'historique du compte de l'utilisateur.

- Page Devis

Un formulaire de devis est mis à disposition pour permettre aux utilisateurs de nous contacter concernant la tarification de nos services de transport.

- Page Contact

Un formulaire de contact sera disponible pour permettre aux utilisateurs de nous contacter rapidement et facilement.

- Page Inscription

Cette page permet aux nouveaux utilisateurs de créer un compte en fournissant leurs informations personnelles de base.

Après l'inscription, les utilisateurs auront accès à l'ensemble des fonctionnalités du site.

- Page Connexion

Les utilisateurs enregistrés se connectent ici en saisissant leur adresse e-mail et leur mot de passe. En cas d'oubli du mot de passe, les utilisateurs peuvent utiliser la fonction "Mot de passe oublié" pour réinitialiser leur mot de passe. Ils recevront un e-mail avec des instructions pour créer un nouveau mot de passe.

Une fois connectés, ils accèdent à leur compte personnel pour effectuer des réservations, voir l'historique des réservations et gérer leur profil.

- Page Profil

Les utilisateurs peuvent accéder à leur profil pour réserver, se déconnecter, supprimer leur compte et accéder à l'historique des réservations.

Architecture du site

Le site a été élaboré en prenant en compte une accessibilité optimale à travers divers navigateurs web, aussi bien sur les ordinateurs que sur les appareils mobiles. La conception repose sur l'utilisation de différentes technologies et langages de programmation, notamment :

- HTML 5 (HyperText Markup Language)

Pour la structuration des pages, garantissant une organisation efficace de l'ensemble du contenu.

- CSS 3 (Cascading Style Sheets)

Pour la mise en page, offrant une présentation esthétique et cohérente.

- JavaScript

Pour améliorer l'interaction et la convivialité de l'interface utilisateur, créant ainsi une expérience utilisateur enrichie.

- PHP (Hypertext Preprocessor)

Pour le développement de la partie backend, permettant de gérer les fonctionnalités essentielles du site.

- SQL (Structured Query Language)

Pour administrer la base de données, assurant ainsi la gestion efficace des données du site.

Environnement de développement

- **Visual Studio Code**

Visual Studio Code est un éditeur de code prenant en charge une grande variété de langages de développement.

- **XAMPP**

Logiciel permettant de simuler un serveur sur un ordinateur avec les modules Apache et MySQL pour la création et la gestion de bases de données.

Spécification des fonctionnalités

Le site propose une gamme de fonctionnalités clés, notamment l'inscription, la connexion, la réinitialisation du mot de passe, l'affichage de l'historique des réservations, la déconnexion et la suppression du compte.

Pour implémenter ces fonctionnalités, un modèle CRUD (Create, Read, Update, Delete) est utilisé pour gérer efficacement les opérations sur la base de données. Voici comment chaque opération CRUD correspond aux fonctionnalités spécifiques du site :

1. **Create** (Créer) : L'opération de création est utilisée lors de l'inscription des utilisateurs. Elle permet d'ajouter de nouvelles informations d'utilisateur à la base de données, telles que l'adresse e-mail, le mot de passe et d'autres détails pertinents.
2. **Read** (Lire) : L'opération de lecture est mise en œuvre pour afficher l'historique des réservations et permet aux utilisateurs de consulter les informations stockées dans la base de données. Cela peut inclure des détails sur les réservations passées, les factures, etc.
3. **Update** (Mettre à jour) : Les utilisateurs peuvent modifier leur mot de passe à l'aide de l'opération de mise à jour. Cela intervient notamment lors de la réinitialisation du mot de passe. Lorsque l'utilisateur souhaite changer son mot de passe, cette opération modifie l'entrée correspondante dans la base de données.
4. **Delete** (Supprimer) : L'opération de suppression est utilisée lorsque les utilisateurs souhaitent supprimer leur compte. Elle supprime toutes les informations associées à un utilisateur de la base de données, y compris ses réservations passées.

L'utilisation du modèle CRUD permet de garantir que les données sont gérées de manière efficace et sécurisée, tout en offrant aux utilisateurs une expérience fluide et fiable sur le site.

Spécifications techniques du projet, élaborées par le candidat, y compris pour la sécurité et le web mobile

Maquettage du site

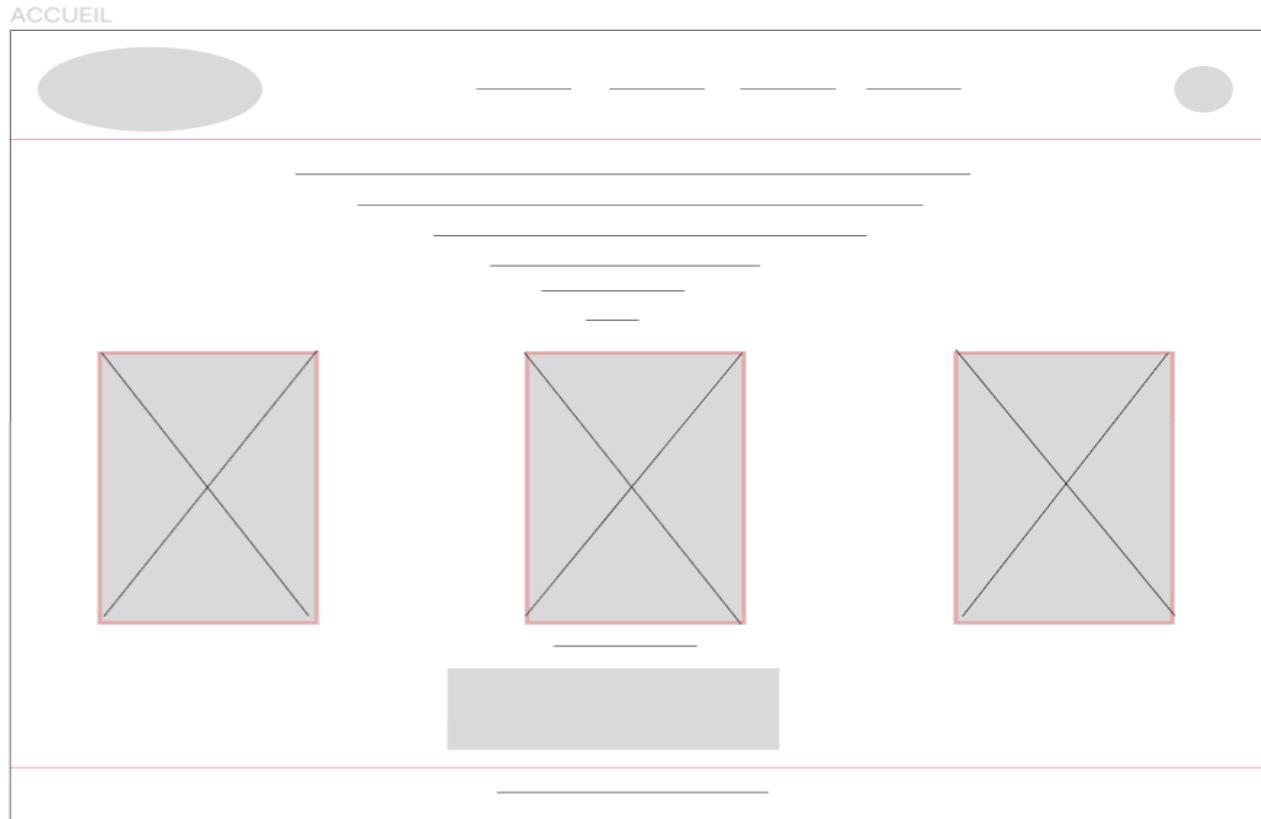
Pour suivre les principes du maquettage de manière rigoureuse, j'ai commencé par analyser la structure du site, en identifiant les pages nécessaires et en élaborant leur apparence.

Ensuite, j'ai employé l'outil Figma pour créer une première ébauche visuelle du site internet, en prenant en compte la palette de couleurs et en déterminant l'emplacement des divers éléments.

J'ai veillé à ce que la maquette soit bien organisée et ergonomique, en prenant en compte l'expérience de l'utilisateur. J'ai également envisagé où placer les menus, les boutons d'interaction et d'autres éléments pour optimiser la navigation.

L'utilisation de Figma m'a permis de concevoir des zonings, des wireframes et des prototypes de maquettes pour chaque page, ce qui s'est révélé utile pour visualiser le rendu final.

Zoning page Accueil :



Wireframe page Accueil :

ACCUEIL

LOGO

ACCUEIL RESERVATION DEVIS CONTACT PROFIL

Besoin d'un taxi ?
Un taxi pour tous vos déplacements
Faites-nous confiance pour vous conduire ou venir vous chercher aux aéroports et aux gares.
Disposant d'un taxi conventionné par la CPAM, nous prenons aussi en main les transports médicaux.
Vous avez besoin d'un service de taxi pour les courses ou les affaires quotidiennes ?
Nous sommes également à votre disposition.

TRANSPORT MEDICAL

TRANSPORT PERSONNEL

TRANSPORT PROFESSIONNEL

Prévisions Météo

Copyright © 2023. Projet Zahra.

Wireframe page Accueil :

ACCUEIL


ZINE TAXI
Transport avec chauffeur

ACCUEIL RESERVATION DEVIS CONTACT PROFIL

Besoin d'un taxi ?
Un taxi pour tous vos déplacements
Faites-nous confiance pour vous conduire ou venir vous chercher aux aéroports et aux gares.
Disposant d'un taxi conventionné par la CPAM, nous prenons aussi en main les transports médicaux.
Vous avez besoin d'un service de taxi pour les courses ou les affaires quotidiennes ?
Nous sommes également à votre disposition.

 TRANSPORT MEDICAL

 TRANSPORT PERSONNEL

 TRANSPORT PROFESSIONNEL

Prévisions Météo

Entrez une ville
Obtenir les prévisions

Prévisions
aucune ville choisie

Copyright © 2023. Projet Zahra.

Réalisations du candidat comportant les extraits de code les plus significatifs et en les argumentant, y compris pour la sécurité et le web mobile

Réaliser une interface utilisateur web statique et adaptable

Après avoir achevé la phase de maquettage, j'ai développé les pages de mon site en utilisant HTML et CSS. J'ai veillé à les rendre adaptatives, de manière à garantir leur affichage optimal sur une variété de dispositifs, qu'il s'agisse d'ordinateurs, de tablettes ou de téléphones mobiles.

Page d'accueil

HTML :

Ma page d'accueil comprend trois parties principales. Tout d'abord, une section de présentation met en avant les services variés de notre service de taxi, y compris le transport médical et les déplacements aux aéroports. Ensuite, une div regroupant les images avec des liens invitant les visiteurs à réserver pour différents types de trajets. Enfin, une div dédiée à la météo permet aux utilisateurs de vérifier les prévisions météorologiques pour une ville spécifique à l'aide d'un formulaire interactif.

```
<main class="accueil-main">
    <section class="presentation">
        <h1>Besoin d'un taxi ?</h1>
        <article>
            <h2>Un taxi pour tous vos déplacements</h2>
            <p> Faites-nous confiance pour vous conduire ou venir vous chercher aux aéroports et aux gares. </p>
            <p>Disposant d'un taxi conventionné par la CPAM, nous prenons aussi en main les transports médicaux. </p>
            <p> Vous avez besoin d'un service de taxi pour les courses ou les affaires quotidiennes ? Nous sommes également à votre disposition.</p>
        </article>
    </section>
    <div class="pictures">
        <figure class="snip1521">
            
            <figcaption>
                <h3 class="title">Transport médical</h3>
                <h3 class="hover">Réserver</h3>
            </figcaption>
            <a href="contact.php"></a>
        </figure>
        <figure class="snip1521">
            
            <figcaption>
                <h3 class="title">Transport personnel</h3>
                <h3 class="hover">Réserver</h3>
            </figcaption>
            <a href="contact.php"></a>
        </figure>
        <figure class="snip1521">
            
            <figcaption>
                <h3 class="title">Transport professionnel</h3>
                <h3 class="hover">Réserver</h3>
            </figcaption>
            <a href="medical.php"></a>
        </figure>
    </div>
    <div class="meteo">
        <h1>Prévisions Météo </h1>
        <form class="weather">
            <input type="text" class="weather__city" autofocus required placeholder="Entrez une ville">
            <input type="submit" class="weather__validate" value="Obtenir les prévisions">
            <h3 class="weather__sub-title">Prévisions</h3>
            <p class="weather__forecast">aucune ville choisie</p>
        </form>
    </div>
</main>
```

CSS :

J'ai conçu un agencement flexible pour les images, stylisé le titre principal, centré le texte dans la section de présentation avec des dimensions spécifiques, et ajusté la taille des paragraphes pour une lisibilité optimale.



J'ai élaboré une animation pour un conteneur principal avec la classe "**snip1521**". J'ai intégré la police Google Open Sans, appliqué des transitions fluides à ses éléments enfants, stylisé les images avec des effets de survol, et ajusté les styles pour une expérience visuelle attrayante lors du survol.



```

/* Styles pour le titre h3 */
.snip1521 h3 {
    text-transform: uppercase;
    font-size: 0.9em;
    font-weight: 400;
    margin: 0;
    letter-spacing: 2px;
}

/* Styles pour le titre h3 au survol */
.snip1521 h3:hover {
    color: #white;
    font-weight: 800;
}

/* Styles pour les liens à l'intérieur de snip1521 */
.snip1521 a {
    position: absolute; /* Positionnement absolu pour occuper tout l'espace du conteneur parent */
    top: 0; /* Position en haut du conteneur parent */
    bottom: 0; /* Position en bas du conteneur parent */
    left: 0; /* Position à gauche du conteneur parent */
    right: 0; /* Position à droite du conteneur parent */
    z-index: 1; /* Z-index pour contrôler l'empilement des éléments */
}

/* Styles appliqués au survol de snip1521 et de ses éléments enfants */
.snip1521:hover > img,
.snip1521:hover > img {
    opacity: 0; /* Masquage de l'image au survol */
}

.snip1521:hover figcaption,
.snip1521:hover figcaption {
    background-color: #000000; /* Changement de la couleur de fond de la légende en noir au survol */
}

.snip1521:hover .title,
.snip1521:hover .title {
    -webkit-transform: translateY(-150%); /* Translation vers le haut de 150% pour le titre au survol */
    transform: translateY(-150%);
}

.snip1521:hover .hover,
.snip1521:hover .hover {
    -webkit-transform: translateY(-150%); /* Translation vers le haut de 150% pour l'élément avec la classe hover au survol */
    transform: translateY(-150%);
}

/* FIN ANIMATION */

```

Media Queries :

J'ai mis en place des styles spécifiques pour les appareils mobiles et tablettes dans mon code CSS. Pour les écrans de largeur maximale de 600 pixels (mobiles), j'ai ajusté la disposition des images, la taille du texte, et les marges pour une expérience de navigation plus adaptée. Pour les écrans compris entre 601 et 1024 pixels (tablettes), j'ai apporté des modifications supplémentaires pour optimiser la présentation, la taille du texte, et l'agencement des images. Ces ajustements visent à garantir une apparence optimale et une lisibilité accrue sur des dispositifs variés.

```
/* mobile */
@media (max-width: 600px) {
  /* Mobile-specific styles */

  .pictures {
    flex-direction: column;
    align-items: center;
    padding: 10px;
  }
  .snip1521 {
    padding: 5px;
    /* min-width: 350px;
    max-width: 445px; */
    width: 88px;
  }

  .pictures-trans {
    margin-top: 10px;
    /* height: auto;
    max-width: 100%; */
  }

  h1 {
    font-size: large;
  }

  .presentation {
    margin: 20px;
    width: 88px;
    margin-left: 40px;
    padding-bottom: 0;
  }

  .presentation p {
    font-size: medium;
  }
  .meteo{
    width: 88px;
  }
}

@media (min-width: 601px) and (max-width: 1024px) {
  /* Tablet-specific styles */

  .pictures {
    flex-wrap: wrap;
    justify-content: center;
  }

  .pictures-trans {
    margin-top: 10px;
  }

  h1 {
    font-size: large;
  }

  .presentation {
    margin: 30px;
  }

  .presentation p {
    font-size: large;
  }
}
```

Page d'accueil sur un navigateur web :

The screenshot shows the desktop version of the Zine Taxi website. At the top, there is a navigation bar with links for Accueil, Réservation, Devis, Contact, and Profil. The main heading is "Besoin d'un taxi ?" followed by a sub-heading "Un taxi pour tous vos déplacements". Below this, there is a brief description: "Faites-nous confiance pour vous conduire ou venir vous chercher aux aéroports et aux gares. Disposant d'un taxi conventionné par la CPAM, nous prenons aussi en main les transports médicaux. Vous avez besoin d'un service de taxi pour les courses ou les affaires quotidiennes ? Nous sommes également à votre disposition." Three service categories are displayed with images: "TRANSPORT MEDICAL" (a man in a wheelchair being assisted), "TRANSPORT PERSONNEL" (a woman with luggage), and "TRANSPORT PROFESSIONNEL" (a woman working on a laptop in a car). A vertical scrollbar is visible on the right side of the page.

Page d'accueil sur un mobile :

The screenshot shows the mobile version of the Zine Taxi website. It features a header with the logo "ZINE TAXI" and a menu icon. The main content area has a large heading "Besoin d'un taxi ?" and a sub-section "Un taxi pour tous vos déplacements". Below this, there is descriptive text: "Faites-nous confiance pour vous conduire ou venir vous chercher aux aéroports et aux gares. Disposant d'un taxi conventionné par la CPAM, nous prenons aussi en main les transports médicaux. Vous avez besoin d'un service de taxi pour les courses ou les affaires quotidiennes ? Nous sommes également à votre disposition." At the bottom, there is a large image of a man in a wheelchair being assisted, with the word "TRANSPORT" overlaid. A horizontal scrollbar is visible at the bottom of the page.

Page d'accueil sur tablette :

The screenshot shows the tablet version of the Zine Taxi website. It has a similar layout to the mobile version, with the "ZINE TAXI" logo and navigation links at the top. The main content includes the "Besoin d'un taxi ?" heading, the "Un taxi pour tous vos déplacements" section, and descriptive text about medical, personal, and professional transport. It also features the three service categories with their respective images. Additionally, there is a "Prévisions Météo" (Weather Forecasts) section with buttons for "Entrer une ville", "Charger les prévisions", "Prévisions", and "Sauvegarde ville choisie". A copyright notice "Copyright © 2022, l'Projet Zahra." is at the bottom. A vertical scrollbar is visible on the right side of the page.

SEO (Search Engine Optimization)

Le SEO, ou optimisation pour les moteurs de recherche, désigne l'ensemble des techniques et des pratiques visant à améliorer la visibilité et le positionnement d'un site web sur les résultats organiques (non payants) des moteurs de recherche tels que Google, Bing, ou Yahoo.

Fonctionnement du SEO :

- **Recherche de Mots-clés** : Le processus commence par la recherche de mots-clés pertinents pour le contenu du site. Ces mots-clés sont ceux que les utilisateurs sont susceptibles de rechercher lorsqu'ils cherchent des informations liées au sujet du site.
- **Optimisation On-Page** : Il s'agit d'optimiser chaque page du site en fonction des mots-clés sélectionnés. Cela inclut l'optimisation du contenu, des balises HTML, des images, des liens internes, etc.
- **Optimisation Off-Page** : Cette étape concerne les facteurs externes au site qui influent sur son classement. Cela inclut la création de liens de qualité provenant d'autres sites, les signaux sociaux (activité sur les réseaux sociaux), et la gestion de la réputation en ligne.
- **Expérience Utilisateur** : Les moteurs de recherche accordent de plus en plus d'importance à l'expérience utilisateur. Des sites bien structurés, faciles à naviguer, et qui offrent un contenu de qualité sont favorisés.
- **Algorithme des Moteurs de Recherche** : Les moteurs de recherche utilisent des algorithmes complexes pour classer les sites en fonction de leur pertinence par rapport aux mots-clés. Ces algorithmes évoluent constamment, ce qui nécessite une adaptation continue des pratiques SEO.

But du SEO :

Le principal objectif du SEO est d'augmenter la visibilité d'un site sur les moteurs de recherche et d'attirer du trafic organique qualifié. En optimisant un site selon les bonnes pratiques du SEO, on cherche à obtenir un meilleur classement dans les résultats de recherche, ce qui peut conduire à une augmentation du trafic, de la notoriété et, éventuellement, des conversions (ventes, inscriptions, etc.). En résumé, le SEO vise à rendre un site web plus accessible et attrayant pour les moteurs de recherche et, par extension, pour les utilisateurs qui effectuent des recherches en ligne.

Exemples :

- Le **<head>** d'une page HTML est une section qui contient des informations métadonnées sur le document. Ces métadonnées sont cruciales pour le référencement (SEO). Voici quelques balises **<meta>** importantes dans le contexte du SEO :

<meta charset="UTF-8"> :

Cette balise spécifie l'encodage des caractères utilisé par la page. **UTF-8** est largement recommandé car il prend en charge une vaste gamme de caractères, essentiel pour les sites multilingues.

<meta http-equiv="X-UA-Compatible" content="IE=edge"> :

Cette balise est souvent utilisée pour définir le mode de compatibilité d'Internet Explorer. **IE=edge** indique que la dernière version disponible d'IE doit être utilisée, améliorant la compatibilité.

<meta name="viewport" content="width=device-width, initial-scale=1.0"> :

Cette balise est cruciale pour le rendu mobile. Elle indique au navigateur comment ajuster la largeur et l'échelle de la page pour s'adapter à différents appareils, améliorant ainsi l'expérience utilisateur sur les dispositifs mobiles.

```
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- La balise **<title>** dans le **<head>** d'une page HTML est pour le SEO avec plusieurs avantages :

Affichage dans les résultats de recherche : Utilisé comme titre principal.

Impact sur le classement : Un titre bien optimisé améliore le classement.

Identification du contenu : Donne une idée claire du contenu de la page.

Optimisation des mots-clés : Endroit clé pour inclure des mots-clés pertinents.

Réduction du taux de rebond : Un titre convaincant attire des visiteurs

```
<title>Zine Taxi</title>
```

- Le texte alternatif (ALT) dans les images offre des avantages SEO :

Accessibilité : Aide les utilisateurs déficients visuels.

SEO amélioré : Contribue à la visibilité dans les résultats de recherche.

Expérience utilisateur : Fournit une compréhension du contenu visuel manquant.

Optimisation des mots-clés : Renforce la pertinence pour les moteurs de recherche.

```

```

Développer une interface utilisateur web dynamique

Sur la version mobile, vous verrez un icône composé de trois barres horizontales.

Cet icône déclenche un menu déroulant qui cache les liens de navigation afin de libérer de l'espace sur l'écran.

En cliquant sur cet icône l'utilisateur se verra activer la liste déroulante et un changement de couleur de celui-ci, et en le cliquant à nouveau, l'utilisateur la désactive.

Cela permet une gestion pratique des liens de navigation sur les versions mobiles.

HTML :

```
<div class="brand-title">
</div>
<a href="#" class="toggle-button">
  <span class="bar"></span>
  <span class="bar"></span>
  <span class="bar"></span>
</a>
<div class="navbar-links">
  <ul>
    <li><a href="homepage.php">Accueil</a></li>
    <li><a href="inscription.php">Réservation</a></li>
    <li><a href="devis.php">Devis</a></li>
    <li><a href="contact.php">Contact</a></li>
  </ul>
</div>
```

Javascript :

```
// MENU HAMBURGER
// Sélectionne le bouton de menu par sa classe CSS 'toggle-button'
const toggleButton = document.getElementsByClassName('toggle-button')[0];

// Sélectionne la liste des liens de navigation par sa classe CSS 'navbar-links'
const navbarLinks = document.getElementsByClassName('navbar-links')[0];

// Ajoute un écouteur d'événement sur le clic du bouton de menu
toggleButton.addEventListener('click', () => {
  // Active ou désactive la classe "active" sur la liste des liens de navigation
  navbarLinks.classList.toggle('active');
  // Active ou désactive la classe "active" sur le bouton de menu (icône)
  toggleButton.classList.toggle('active');
});
```

Menu déroulant ouvert sur mobile :



CSS :

```
/* bouton de bascule qui est utilisé pour afficher ou masquer la barre de navigation */
.toggle-button {
  display: none; /* pour afficher et masquer le bouton menu */
  flex-direction: column;
  justify-content: center;
  align-items: flex-end;
  cursor: pointer;
  width: 30px;
  height: 30px;
  position: absolute; /* pour qu'il reste fixé */
  top: 50%; /* pour ajuster la position verticale */
  right: 20px; /* pour ajuster la position horizontale */
  transform: translateY(-50%); /* pour déplacer le bouton vers le haut de 50% de sa propre hauteur */
}

/* bouton menu */
.toggle-button .bar {
  width: 100%;
  height: 2px;
  background-color: black;
  margin-bottom: 4px;
  transition: background-color 0.3s ease; /* Animation pour ajouter un effet de transition fluide */
}

/* changement de couleur lorsqu'elles sont actives */
.toggle-button.active .bar {
  background-color: #F57171;
}
```

Création de la base de données

Pour concevoir ma base de données, j'ai débuté par définir les objectifs et les exigences du système, en analysant les besoins des utilisateurs. Pour structurer les données, j'ai créé un dictionnaire des données, une documentation détaillée répertoriant les informations cruciales de chaque élément de chaque table.

Ce dictionnaire inclut le nom, le format, la longueur, le type (élémentaire ou calculé), les règles de calcul et de gestion, ainsi qu'une référence documentaire. Cet outil centralisé est essentiel pour la compréhension et la gestion efficace de la base de données.

| DICTIONNAIRE DE DONNEES ZINE TAXI | | | | | | | |
|--------------------------------------|----------------|----------|-------------------------------------|--------------------------|-----------------|------------------|------------------|
| Nom de la donnée | Format | Longueur | Type | | Règle de calcul | Règle de gestion | Document |
| | | | Elementaire | Calculé | | | |
| INSCRIPTION | | | | | | | |
| Id_inscription | Numérique | | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page inscription |
| Nom | Alphabétique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page inscription |
| Prénom | Alphabétique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page inscription |
| Email | Alphanumérique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page inscription |
| Pass | Alphanumérique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page inscription |
| DEVIS | | | | | | | |
| Id-devis | Numérique | | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page devis |
| Nom | Alphabétique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page devis |
| Prénom | Alphabétique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page devis |
| Numéro de téléphone | Numérique | 20 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page devis |
| Email | Alphanumérique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page devis |
| Type | Alphanumérique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page devis |
| Départ | Alphanumérique | 300 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page devis |
| Arrivée | Alphanumérique | 300 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page devis |
| Date | Numérique | 8 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page devis |
| Heure | Numérique | 4 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page devis |
| RESERVATION MEDICAL | | | | | | | |
| Id-medical | Numérique | | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page médical |
| Nom | Alphabétique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page médical |
| Prénom | Alphabétique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page médical |
| Numéro de téléphone | Numérique | 20 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page médical |
| Email | Alphanumérique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page médical |
| Type | Alphanumérique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page médical |
| Départ | Alphanumérique | 300 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page médical |
| Arrivée | Alphanumérique | 300 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page médical |
| Date | Numérique | 8 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page médical |
| Heure | Numérique | 4 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page médical |
| Bon de transport | File | | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page médical |
| Attestation carte vitale | File | | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | Page médical |

| <u>RESERVATION PRO</u> | | | | | |
|-------------------------------------|----------------|------|-------------------------------------|--------------------------|----------------------------|
| Id-pro | Numérique | | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page pro |
| Nom | Alphabétique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page pro |
| Prénom | Alphabétique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page pro |
| Numéro de téléphone | Numérique | 20 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page pro |
| Email | Alphanumérique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page pro |
| Type | Alphanumérique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page pro |
| Départ | Alphanumérique | 300 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page pro |
| Arrivée | Alphanumérique | 300 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page pro |
| Date | Numérique | 8 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page pro |
| Heure | Numérique | 4 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page pro |
| Nom-paiement | Alphabétique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page pro |
| Numéro bancaire | Numérique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page pro |
| Date d'expiration | Numérique | 6 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page pro |
| Cryptogramme | Numérique | 3 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page pro |
| <u>RESERVATION PERSO</u> | | | | | |
| Id-perso | Numérique | | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page perso |
| Nom | Alphabétique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page perso |
| Prénom | Alphabétique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page perso |
| Numéro de téléphone | Numérique | 20 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page perso |
| Email | Alphanumérique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page perso |
| Aller simple | | | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page perso |
| Aller retour | | | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page perso |
| Départ | Alphanumérique | 300 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page perso |
| Arrivée | Alphanumérique | 300 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page perso |
| Date | Numérique | 8 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page perso |
| Heure | Numérique | 4 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page perso |
| Nom-paiement | Alphabétique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page perso |
| Numéro bancaire | Numérique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page perso |
| Date d'expiration | Numérique | 6 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page perso |
| Cryptogramme | Numérique | 3 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page perso |
| <u>CONTACT</u> | | | | | |
| Id-contact | Numérique | | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page contact |
| Nom | Alphabétique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page contact |
| Prénom | Alphabétique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page contact |
| Numéro de téléphone | Numérique | 20 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page contact |
| Email | Alphanumérique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page contact |
| Message | Alphanumérique | 1000 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page contact |
| <u>OPTIONS SELECTIONNEES</u> | | | | | |
| Id_option_selectionne | Numérique | | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page options selectionnees |
| Option selectionnee | Alphanumérique | 50 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Page options selectionnees |

En utilisant des outils de modélisation comme Drawio, j'ai créé un **Modèle Conceptuel des Données (MCD)**, un outil qu'on utilise pour comprendre et représenter de manière graphique la structure d'une base de données mettant en lumière les entités, leurs attributs, et les relations entre elles.

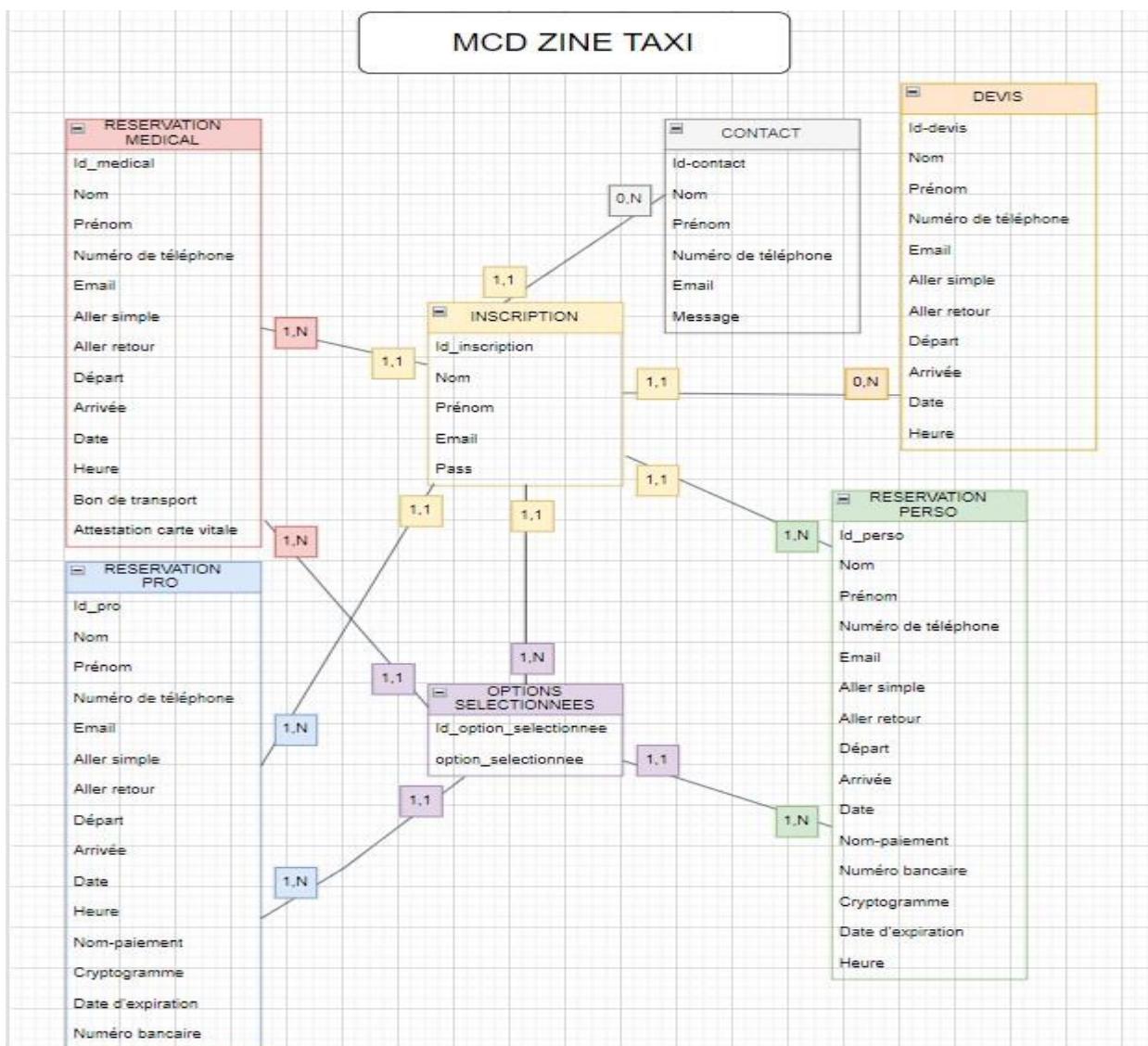
Inscription (Utilisateur) : Chaque utilisateur est unique et s'inscrit une seule fois. Les informations d'un utilisateur sont uniques à lui-même.

Réservation médicale, Réservation professionnelle, Réservation personnelle : Ces tables ont une relation "1,N" avec la table Inscription. C'est-à-dire qu'un utilisateur peut faire une à plusieurs réservations de chaque type. De plus, chaque réservation est faite par un utilisateur, donc de leur côté, la cardinalité est de "1,1".

Options sélectionnées : Cette table a également une relation "1,N" avec la table Inscription car un utilisateur peut sélectionner plusieurs options, mais chaque option appartient à un seul utilisateur et à une réservation donc la cardinalité est "1,1". La cardinalité est "1,N" entre les réservations et les options sélectionnées.

Devis : La table Devis a une relation "0,N" avec la table Inscription car un utilisateur peut demander plusieurs devis. Un devis est demandé par un utilisateur, donc de son côté, la cardinalité est de "1,1".

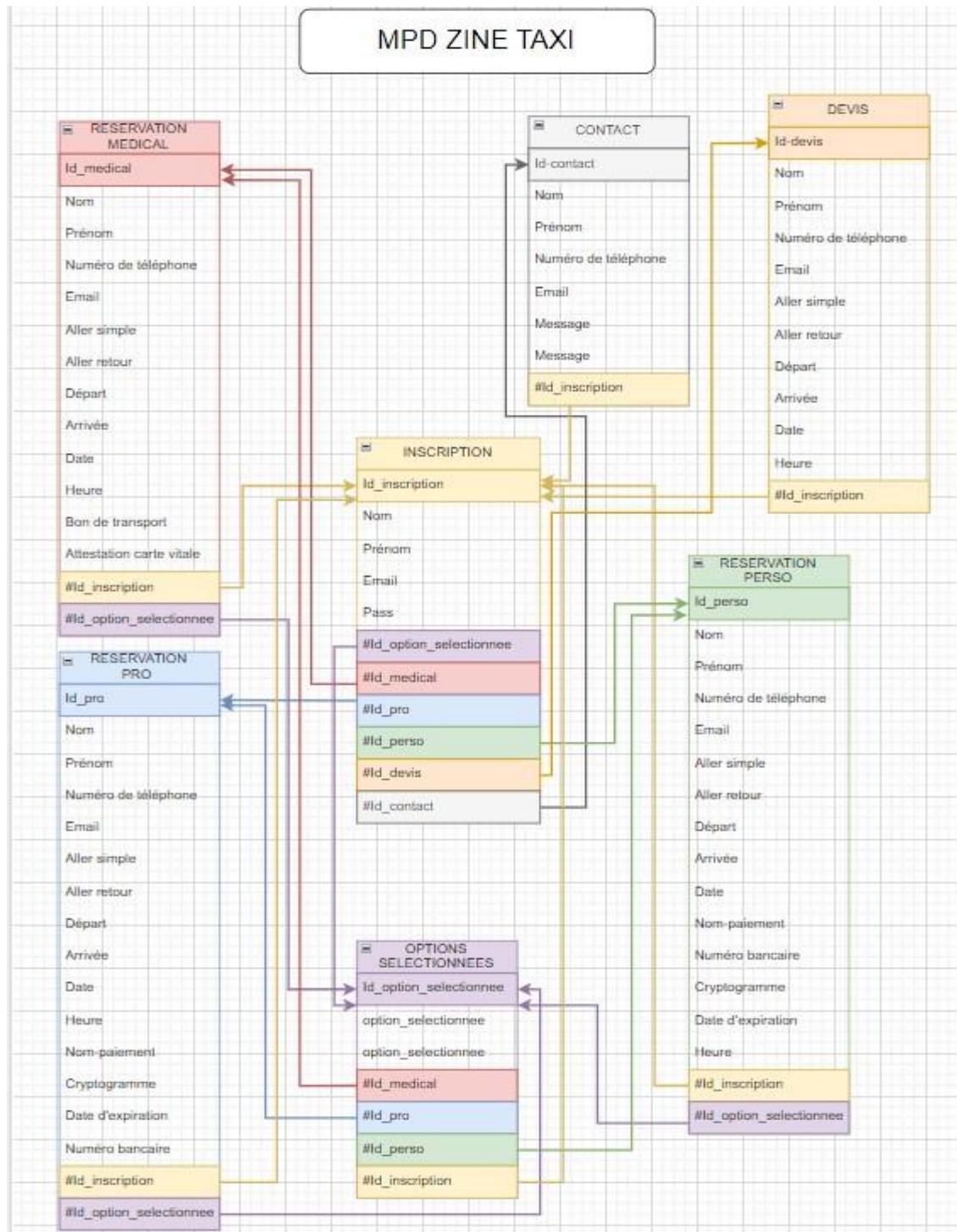
Contact : La table Contact a une relation "0,N" avec la table Inscription car un utilisateur peut envoyer plusieurs messages de contact. Un message de contact est envoyé par un utilisateur, donc de son côté, la cardinalité serait de "1,1".



Pour passer à la mise en œuvre, j'ai transformé le MCD en un **Modèle Physique des Données (MPD)**, une représentation technique de la structure de la base de données, détaillant les aspects tels que les clés primaires, les clés étrangères et les contraintes.

Dans chaque table du MPD, la clé étrangère "Id_inscription" est présente, établissant une liaison avec la table "Inscription" qui correspond à l'utilisateur. Cette relation garantit l'intégrité référentielle, assurant que chaque enregistrement dans une table est lié à un enregistrement existant dans la table "Inscription".

De plus, dans la table "Options sélectionnées", les clés étrangères "id_medical", "id_pro" et "id_perso" sont incluses, établissant des liens avec les tables respectives pour définir le type de réservation associé.



J'ai choisi de mettre en lumière le système de manière plus large à travers une représentation UML (**Unified Modeling Language**). Pour offrir une image précise des interactions, j'ai opté pour le diagramme séquentiel. Ce choix a permis de tracer de manière chronologique le flux d'opérations au sein de la base de données.

Inscription :

Utilisateur : Remplit le formulaire avec ses informations.

Système : Valide les données du formulaire, vérifie la disponibilité de l'adresse e-mail dans la base de données.

Base de données : Recherche l'adresse e-mail dans la table d'inscription.

Si l'adresse e-mail est disponible :

Système : Crée un compte utilisateur avec les informations fournies.

Base de données : Enregistre les données du nouvel utilisateur dans la table d'inscription, génère un identifiant unique (Id_inscription) pour le nouvel utilisateur.

Système : Envoie un message de confirmation de l'inscription à l'utilisateur et le redirige vers la page connexion.

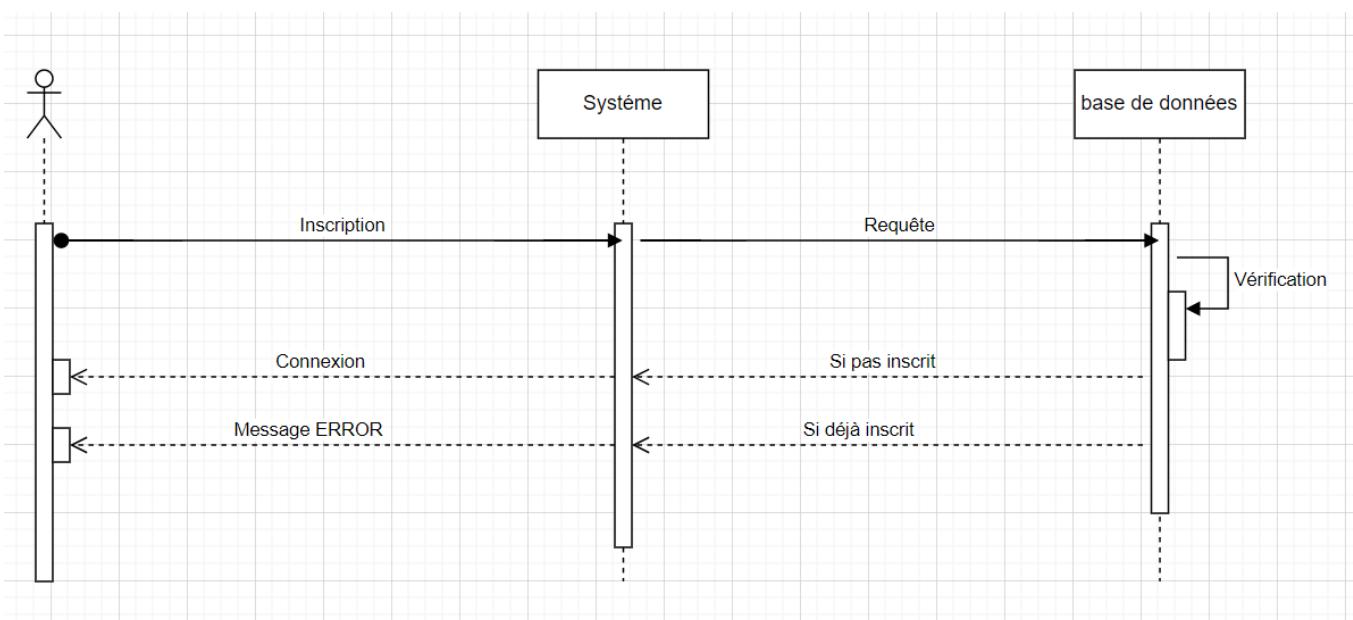
Utilisateur : Reçoit le message de confirmation et il est redirigé vers la page connexion.

Si l'adresse e-mail existe déjà :

Système : Affiche un message d'erreur à l'utilisateur indiquant que l'adresse e-mail est déjà utilisée.

Utilisateur : Révise le formulaire et soumet à nouveau les informations.

Le processus se répète jusqu'à ce qu'une adresse e-mail unique soit fournie.



Connexion :

Utilisateur : Saisit son adresse e-mail et son mot de passe dans le formulaire de connexion.

Système : Vérifie la validité des informations fournies par l'utilisateur.

Base de données : Recherche l'adresse e-mail dans la table d'inscription.

Si l'adresse e-mail existe :

Base de données : Compare le mot de passe saisi avec celui enregistré pour l'adresse e-mail donnée.

Si les informations sont valides :

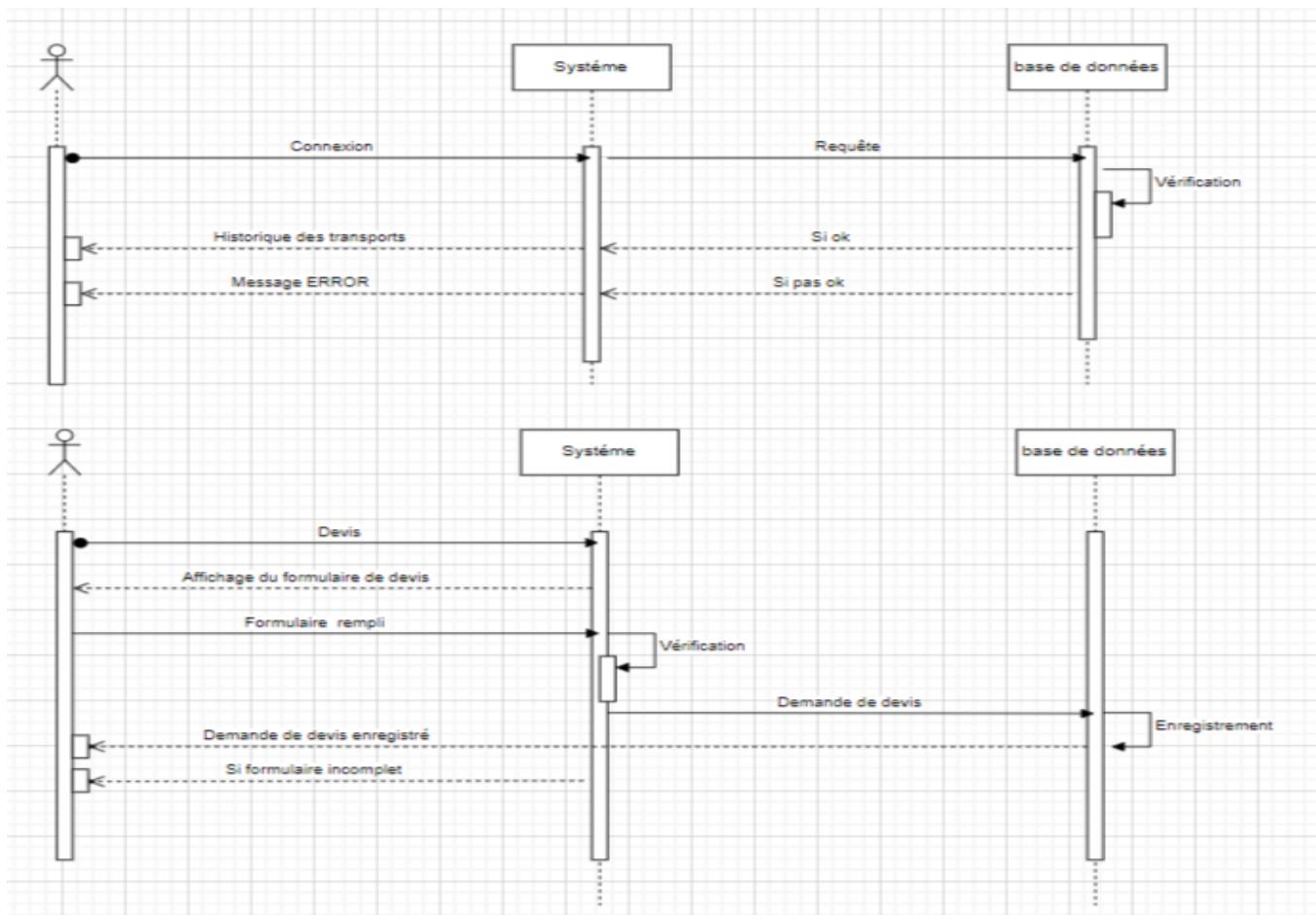
Système : Authentifie l'utilisateur et lui donne accès à l'application.

Utilisateur : Accède à son espace personnel.

Si l'adresse e-mail n'existe pas ou le mot de passe ne correspond pas :

Système : Affiche un message d'erreur indiquant que les informations de connexion sont incorrectes.

Utilisateur : Peut réessayer en fournissant des informations valides ou choisir l'option de récupération de mot de passe si nécessaire.



Devis :

Utilisateur : Remplit le formulaire de demande de devis avec les détails nécessaires.

Système : Valide les données du formulaire, vérifie si toutes les informations requises sont fournies.

Si le formulaire est incomplet :

Système : Indique à l'utilisateur les champs manquants et l'invite à les remplir.

Si tout est correct :

Base de données : Stocke les informations de la demande dans la table Devis, génère un identifiant unique (Id_devis) pour la demande.

Système : Envoie une confirmation de la demande de devis à l'utilisateur.

Contact :

Utilisateur : Remplit le formulaire de demande de contact avec les informations nécessaires.

Système : Valide les données du formulaire, vérifie si toutes les informations requises sont fournies.

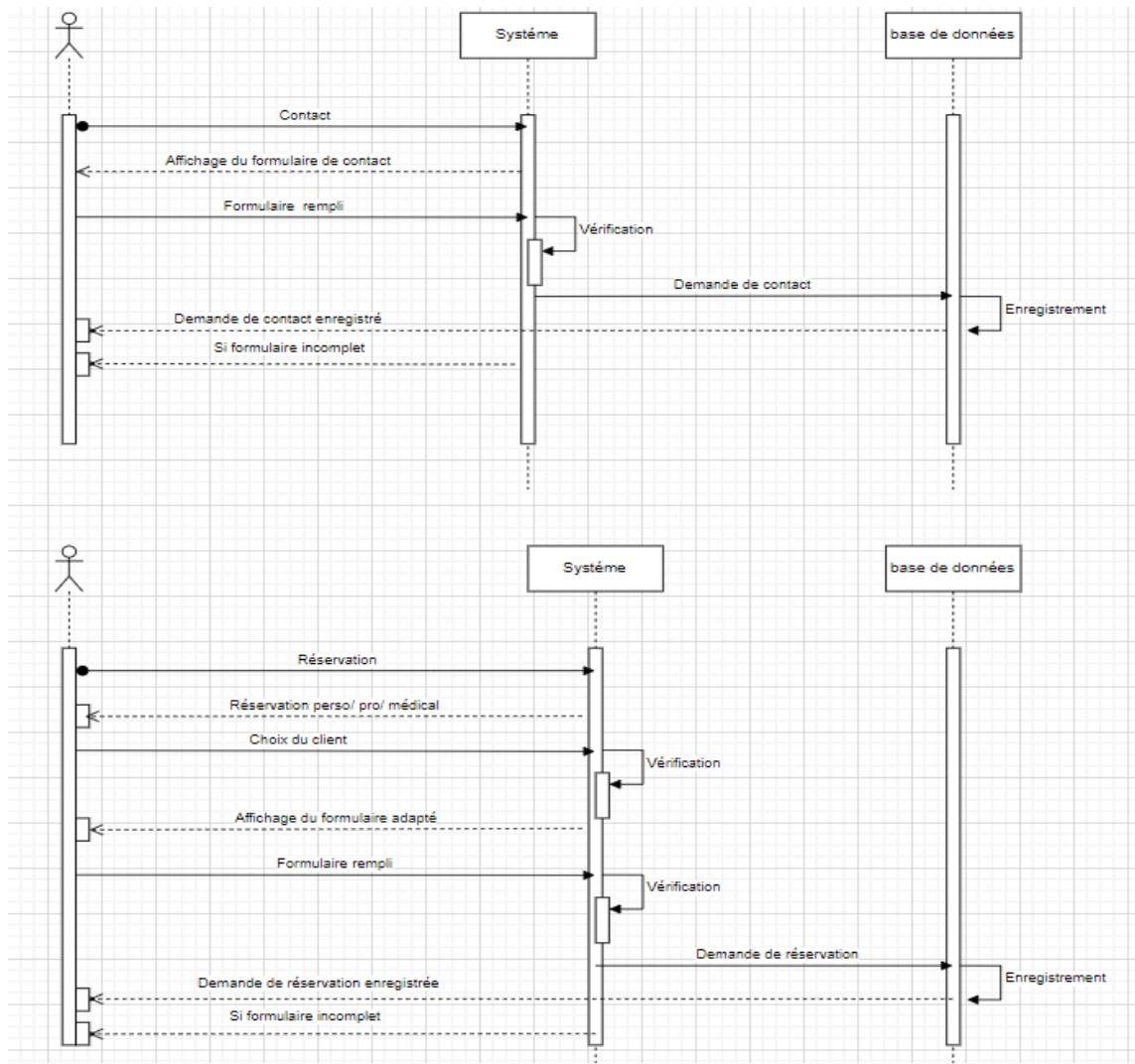
Si le formulaire est incomplet :

Système : Indique à l'utilisateur les champs manquants et l'invite à les remplir.

Si tout est correct :

Base de données : Stocke les informations de la demande dans la table Contact, génère un identifiant unique (Id_contact) pour la demande.

Système : Envoie une confirmation de la demande de contact à l'utilisateur.



Réservation :

Utilisateur : Choisi le type de transport (médical, professionnel, personnel).

Système : Valide le choix du type de transport.

Utilisateur : Accède au formulaire de réservation correspondant.

Système : Vérifie si toutes les informations requises sont fournies dans le formulaire de réservation

Si le formulaire est incomplet :

Système : Indique à l'utilisateur les champs manquants et l'invite à les remplir.

Si tout est correct :

Base de données : Stocke les informations de la réservation dans la table correspondante (reservation_medicale, reservation_pro, reservation_perso), génère un identifiant unique.

Système : Envoie une confirmation de la réservation à l'utilisateur.

Pour la dernière étape de la création de ma base de données, j'ai opté pour le Système de Gestion de Base de Données (SGBD) le plus approprié, avec XAMPP MySQL en tant qu'outil principal. J'ai concrétisé la mise en œuvre de la base de données en suivant rigoureusement le Modèle Physique de Données (MPD).

The screenshot shows the phpMyAdmin interface for the 'zinc_taxi' database. The top navigation bar includes tabs for Structure, SQL, Rechercher, Requête, Exporter, Importer, Opérations, Priviléges, Procédures stockées, Événements, Déclencheurs, and Plus. Below the navigation bar is a search bar labeled 'Contenant le mot:' with a placeholder 'Rechercher'. A 'Filtres' button is also present. The main area displays a table of 8 tables with their respective details:

| Table | Action | Lignes | Type | Interclassement | Taille | Perte |
|-----------------------|---|--------|--------|--------------------|-----------|-------|
| codes | Parcourir Structure Rechercher Insérer Vider Supprimer | 79 | InnoDB | utf8_general_ci | 16,0 kio | - |
| contact | Parcourir Structure Rechercher Insérer Vider Supprimer | 8 | InnoDB | utf8_general_ci | 48,0 kio | - |
| devis | Parcourir Structure Rechercher Insérer Vider Supprimer | 4 | InnoDB | utf8_general_ci | 48,0 kio | - |
| inscription | Parcourir Structure Rechercher Insérer Vider Supprimer | 12 | InnoDB | utf8_general_ci | 80,0 kio | - |
| options_selectionnees | Parcourir Structure Rechercher Insérer Vider Supprimer | 19 | InnoDB | utf8_general_ci | 80,0 kio | - |
| reservation_medcale | Parcourir Structure Rechercher Insérer Vider Supprimer | 1 | InnoDB | utf8_general_ci | 64,0 kio | - |
| reservation_perso | Parcourir Structure Rechercher Insérer Vider Supprimer | 3 | InnoDB | utf8_general_ci | 64,0 kio | - |
| reservation_pro | Parcourir Structure Rechercher Insérer Vider Supprimer | 3 | InnoDB | utf8_general_ci | 80,0 kio | - |
| 8 tables | Somme | 129 | InnoDB | utf8mb4_general_ci | 480,0 kio | 0 o |

At the bottom left, there are buttons for 'Tout cocher' and 'Avec la sélection :'. On the right, there is a dropdown menu.

The screenshot shows the phpMyAdmin interface for the 'inscription' table. The top navigation bar includes tabs for Parcourir, Structure, SQL, Rechercher, Insérer, Exporter, Importer, Priviléges, Opérations, and Déclencheurs. Below the navigation bar is a code editor containing the SQL command: `ALTER TABLE `inscription` CHANGE `id_contact` `id_contact` INT(11) NULL;`. There are buttons for 'Éditer en ligne', 'Éditer', and 'Créer le code source PHP'.

The main area displays the structure of the 'inscription' table:

| # | Nom | Type | Interclassement | Attributs | Null | Valeur par défaut | Commentaires | Extra | Action |
|----|-------------------------------|--------------|-----------------|-----------|------|-------------------|--------------|----------------|-------------------------|
| 1 | id_inscription | int(11) | | | Non | Aucun(e) | | AUTO_INCREMENT | Modifier Supprimer Plus |
| 2 | nom | varchar(50) | utf8_general_ci | | Oui | NULL | | | Modifier Supprimer Plus |
| 3 | prenom | varchar(50) | utf8_general_ci | | Oui | NULL | | | Modifier Supprimer Plus |
| 4 | email | varchar(50) | utf8_general_ci | | Oui | NULL | | | Modifier Supprimer Plus |
| 5 | pass | varchar(255) | utf8_general_ci | | Oui | NULL | | | Modifier Supprimer Plus |
| 6 | active | tinyint(4) | | | Oui | 1 | | | Modifier Supprimer Plus |
| 7 | id_option_selectionnee | int(11) | | | Oui | NULL | | | Modifier Supprimer Plus |
| 8 | id_medical | int(11) | | | Oui | NULL | | | Modifier Supprimer Plus |
| 9 | id_pro | int(11) | | | Oui | NULL | | | Modifier Supprimer Plus |
| 10 | id_perso | int(11) | | | Oui | NULL | | | Modifier Supprimer Plus |
| 11 | id_contact | int(11) | | | Oui | NULL | | | Modifier Supprimer Plus |
| 12 | id_devis | int(11) | | | Oui | NULL | | | Modifier Supprimer Plus |

At the bottom left, there are buttons for 'Tout cocher' and 'Avec la sélection :'. Below the table, there are buttons for 'Parcourir', 'Modifier', 'Supprimer', 'Primaire', 'Unique', 'Index', 'Spatial', and 'Texte entier'.

Ce code SQL définit deux tables, contact et devis, ainsi que les données qu'elles contiennent :

Table contact :

id_contact : identifiant du contact.

nom, prenom : nom et prénom du contact.

phone, email : numéro de téléphone et adresse e-mail du contact.

message : message envoyé par le contact.

envoyé_le : date et heure d'envoi du message.

id_inscription : identifiant lié à une inscription (si utilisateur inscrit (facultatif)).

Table devis :

id_devis : identifiant du devis.

type : type de devis.

nom, prenom : nom et prénom de la personne demandant le devis.

phone, email : numéro de téléphone et adresse e-mail de la personne.

typeAR : type de trajet : aller-simple ou aller-retour.

depart, arrivee : lieu de départ et d'arrivée pour le devis.

date, heure : date et heure de la demande du devis.

message : message supplémentaire pour la demande de devis.

id_inscription : identifiant lié à une inscription (si utilisateur inscrit (facultatif)).

Les données insérées dans ces tables sont des exemples de contacts et de demandes de devis avec différentes informations associées.

```
-- Structure de la table 'contact'
-- 
CREATE TABLE `contact` (
  `id_contact` int(11) NOT NULL,
  `nom` varchar(50) NOT NULL,
  `prenom` varchar(50) NOT NULL,
  `phone` varchar(20) NOT NULL,
  `email` varchar(50) NOT NULL,
  `message` varchar(1000) NOT NULL,
  `envoyé_le` timestamp NOT NULL DEFAULT current_timestamp(),
  `id_inscription` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;

-- Déchargement des données de la table 'contact'
-- 

INSERT INTO `contact`(`id_contact`, `nom`, `prenom`, `phone`, `email`, `message`, `envoyé_le`, `id_inscription`) VALUES
(8, 'pascal', 'pascal', '0766876689', 'Dm@yahoo.com', '?????', '2023-11-13 13:24:29', NULL),
(7, 'ELH', 'zahra', '0766876689', 'zahra.elhasnaoui92@gmail.com', 'Hello', '2023-11-13 13:24:29', NULL),
(8, 'ELH', 'zahra', '0766876689', 'zahra.elhasnaoui92@gmail.com', 'UPDATE - Changement de Mot de Passe', '2023-11-13 13:25:02', NULL),
(8, 'ali', 'ali', '0766666666', 'ali@outlook.com', 'CM#039;EST MOI', '2023-11-14 22:16:01', NULL);

-- 
-- Structure de la table 'devis'
-- 
CREATE TABLE `devis` (
  `id_devis` int(11) NOT NULL,
  `type` varchar(20) NOT NULL,
  `nom` varchar(50) NOT NULL,
  `prenom` varchar(50) NOT NULL,
  `phone` varchar(20) NOT NULL,
  `email` varchar(50) NOT NULL,
  `typeAR` varchar(20) NOT NULL DEFAULT '',
  `depart` varchar(300) NOT NULL,
  `arrivee` varchar(300) NOT NULL,
  `date` date NOT NULL,
  `heure` time(4) NOT NULL,
  `message` varchar(1000) NOT NULL,
  `id_inscription` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;

-- Déchargement des données de la table 'devis'
-- 

INSERT INTO `devis`(`id_devis`, `type`, `nom`, `prenom`, `phone`, `email`, `typeAR`, `depart`, `arrivee`, `date`, `heure`, `message`, `id_inscription`) VALUES
(2, 'sabrina', 'sabrina', '0033612457845', 'sabrina@live.fr', 'aller_simple', '12 rue de louest 03000 cannes', '2023-07-28', '16:21:00.0000', 'Bonjour vous ', NULL),
(3, 'sophia', 'sophia', '33612457845', 'sophia@gmail.com', 'aller_retour', '98 rue de la pierre 92000 nanterre', '98 rue de la pierre 92000 nanterre', '2023-07-26', '17:00:00.0000', 'j#039;aimerais un devis ', NULL),
(4, 'ola', 'ola', '0766875896', 'zahr@gmail.com', 'aller_retour', 'nogent sur oise', '60188', '2023-08-15', '15:55:00.0000', 'ola', NULL),
(5, 'Dauda', 'Mubby', '0766876689', 'Muby@yahoo.com', 'aller_simple', '37 rue de la liberte 60190 nogent', '37 rue de la liberte 60190 nogent', '2023-11-24', '19:03:00.0000', 'Bonjour, j#039;aimerais connaitre le prix d#039;un transport ', NULL),
(7, 'medical', 'ELH', 'zahra', '0766666666', 'zahra.elhasnaoui92@gmail.com', 'aller_simple', '12 rue de la ville 77000 Melun', '12 rue de la ville 77000 Melun', '2023-12-10', '16:31:00.0000', 'hiiiiiii', NULL),
(8, ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', NULL);
```

Architecture et Structure : Développement Modulaire et Gestion des Données dans mon Application Web

Pour le développement de mon application web, j'ai utilisé une approche basée sur l'inclusion de fichiers et le traitement direct. Voici comment ça fonctionne :

Pages individuelles :

Chaque page de mon site est un fichier PHP distinct. Ces fichiers contiennent à la fois la structure HTML de la page et toute la logique PHP nécessaire pour cette page. Cela permet d'avoir une vision claire de ce qui se passe sur chaque page et de garder la logique spécifique à cette page au même endroit.

Inclusion de fichiers :

Pour les éléments répétitifs, comme l'en-tête et le pied de page, j'ai créé des fichiers spécifiques que j'inclus dans chaque page. Cela réduit la duplication du code et facilite la maintenance, car je peux modifier ces éléments en un seul endroit et les changements seront répercutés sur toutes les pages.

```
1  <?php  
2  include ("header.php");  
3  ?>
```

```
63  <?php  
64  include ("footer.php");  
65  ?>
```

Traitement direct des actions :

Plutôt que d'adopter une couche de contrôleur distincte, j'ai choisi de traiter directement les actions de l'utilisateur dans des fichiers de traitement dédiés. Lorsqu'un utilisateur soumet un formulaire, les données sont envoyées à un fichier de traitement spécifique grâce à la méthode POST et à l'attribut

```
<form class ="form-in" action="traitement_inscrip.php" method="POST">
```

"action" dans la balise <form>.

Fichiers constants pour la connexion à la base de données :

J'ai intégré un fichier **constants.bdd.php** pour stocker les informations de connexion à la base de données, garantissant ainsi une gestion centralisée des paramètres de connexion.

```
1  <?php  
2  include("./bdd/constants.bdd.php");
```

Développer les composants d'accès aux données

J'ai utilisé deux approches différentes pour gérer la connexion à la base de données dans mon code.

La première approche utilise une structure switch pour définir les informations de connexion en fonction de l'environnement du serveur. C'est utile si tu as différents environnements qui nécessitent des détails de connexion différents.

Activation des Messages d'Erreurs :

J'ai configuré le script PHP pour améliorer la gestion des erreurs pendant son exécution.

Configuration des Messages d'Erreurs :

error_reporting(E_ALL): Cette instruction permet de voir toutes les erreurs PHP qui pourraient survenir lors de l'exécution du script.

ini_set('display_errors', '1'): Elle force l'affichage des erreurs directement à l'écran, facilitant ainsi le repérage et la résolution des problèmes.

ini_set('log_errors', '0'): Si cette valeur est fixée à '0', cela désactive l'enregistrement des erreurs dans un fichier. Si changée à '1', les erreurs seront enregistrées dans un fichier.

ini_set('error_log', './'): Cela définit le répertoire où le fichier d'erreur sera créé si l'enregistrement des erreurs est activé.

Connexion à la Base de Données (BDD) :

Détermination des Informations de Connexion :

J'utilise une structure de commutation (**switch**) basée sur le serveur (**\$_SERVER['HTTP_HOST']**) pour déterminer quelles constantes utiliser pour la connexion à la base de données en fonction de l'environnement (localhost ou baobab-svr5).

```
1 <?php
2 // activation des messages d'erreurs
3 error_reporting(E_ALL); // permet de voir toutes les erreurs PHP qui se produisent lorsque ton script s'exécute.
4 ini_set('display_errors', '1'); //force l'affichage des erreurs à l'écran.
5 ini_set('log_errors', '0'); //quand elle est fixée à '0', désactive l'enregistrement des erreurs dans un fichier.
6 // si tu changes cela en '1', les erreurs seront enregistrées dans un fichier.
7 ini_set('error_log', './'); //définit le répertoire où le fichier d'erreur sera créé si log_errors est activé.
8
9 // connexion BDD
10 // pour déterminer quelles informations utiliser pour se connecter à la base de données
11 switch ($_SERVER['HTTP_HOST']) {
12     case 'localhost':
13         //définis des constantes (HOST, PORT, DATA, USER, PASS) avec les informations de connexion correspondantes.
14         define('HOST', 'localhost');
15         define('PORT', 3306);
16         define('DATA', 'zine_taxi');
17         define('USER', 'root');
18         define('PASS', '');
19         break;
20     case 'baobab-svr5': // Fictif
21         define('HOST', 'baobab-svr5');
22         define('DATA', 'baobab-sql5');
23         define('USER', 'baobab-usr1');
24         define('PASS', 'aR5*hgt+7uIopp$');
25         break;
26     default:
27         # code...
28         break;
29 }
?>
```

La deuxième approche, elle, crée effectivement la connexion à la base de données. J'ai utilisé PHP Data Objects (PDO) pour cela. J'ai aussi défini certains attributs de PDO, comme le mode de gestion des erreurs et l'utilisation des vraies requêtes préparées plutôt que de leur émulation.

Connexion à la Base de Données avec PDO :

J'utilise PHP Data Objects (PDO), une extension PHP qui offre une approche orientée objet pour interagir avec des bases de données MySQL.

Informations de Connexion :

\$db_host: L'hôte de la base de données (dans cet exemple, "localhost").

\$db_name: Le nom de la base de données ("zine_taxi").

\$db_user: L'utilisateur de la base de données ("root").

\$db_pass: Le mot de passe de l'utilisateur de la base de données (vide dans cet exemple).

Création de l'Objet PDO :

J'utilise ces informations pour créer un nouvel objet PDO qui représente la connexion à la base de données.

La chaîne de connexion spécifie le type de base de données (MySQL), l'hôte, le nom de la base de données, le jeu de caractères (utf8), et les identifiants d'utilisateur et de mot de passe.

\$pdo = new PDO("mysql:host=\$db_host;dbname=\$db_name;charset=utf8", \$db_user, \$db_pass);

Gestion des Erreurs PDO :

J'ai configuré PDO pour lancer une exception (PDO::ERRMODE_EXCEPTION) chaque fois qu'une erreur survient, facilitant ainsi le repérage et le traitement des problèmes.

\$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

Désactivation de l'Émulation des Requêtes Préparées :

J'ai désactivé l'émulation des requêtes préparées (PDO::ATTR_EMULATE_PREPARES) pour favoriser l'utilisation de vraies requêtes préparées. Cela renforce la sécurité et évite les vulnérabilités liées aux injections SQL.

\$pdo->setAttribute(PDO::ATTR_EMULATE_PREPARES, false);

```
1 <?php
2 // se connecte à une base de données MySQL avec PHP Data Objects (PDO).
3 //Le PDO est une extension de PHP qui fournit une approche orientée objet pour accéder à des bases de données.
4
5 $db_host = "localhost";// l'hôte de la base de données
6 $db_name = "zine_taxi";//le nom de la base de données
7 $db_user = "root";// l'utilisateur de la base de données
8 $db_pass = "";//le mot de passe de l'utilisateur de la base de données.
9
10 // crée un nouvel objet PDO avec les informations de connexion à la base de données
11 $pdo = new PDO("mysql:host=$db_host;dbname=$db_name;charset=utf8", $db_user, $db_pass);
12
13 // Afficher les erreurs PDO
14 //configurer PDO pour lancer une exception chaque fois qu'une erreur se produit.
15 $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
16 //désactive l'émulation des requêtes préparées et permet l'utilisation de requêtes préparées réelles.
17 $pdo->setAttribute(PDO::ATTR_EMULATE_PREPARES, false);
18 ?>
```

Avec ces deux approches combinées, j'ai une gestion flexible et robuste des connexions à ma base de données, qui peut s'adapter à différents environnements.

Après avoir défini les constantes de la base de données, je procède à la gestion des données du formulaire. Le script récupère les informations saisies, les valide en vérifiant leur format, les protège contre les attaques XSS, et gère toute erreur potentielle. Si des erreurs sont détectées, l'utilisateur est redirigé vers le formulaire avec un message d'erreur.

Inclusion des Constantes de la Base de Données :

Le code inclut un fichier **constants.bdd.php** qui contient probablement les informations de connexion à la base de données.

Démarrage de la Session :

session_start() est appelé pour démarrer une session PHP. Les données du formulaire et les erreurs seront stockées dans la session.

Récupération des Données du Formulaire :

Les données du formulaire (nom, prénom, téléphone, email, message) sont récupérées à partir du tableau **\$_POST**.

Validation des Données :

Les données sont validées à l'aide d'expressions régulières et de filtres, et des messages d'erreur sont ajoutés au tableau **\$erreurs** en cas d'échec de validation.

Protection contre les Attaques XSS :

Les données sont traitées avec la fonction **htmlspecialchars** pour se protéger contre les attaques XSS.

Gestion des Erreurs :

Si des erreurs sont détectées, les données et les erreurs sont stockées dans la session, et l'utilisateur est redirigé vers le formulaire avec un message d'erreur.

```

1  <?php
2  include("./bdd/constants.bdd.php");
3  session_start();
4
5 //Récuperer les données du formulaires
6 $nom = isset($_POST['nom']) ? $_POST['nom'] : '';
7 $prenom = isset($_POST['prenom']) ? $_POST['prenom'] : '';
8 $phone = isset($_POST['phone']) ? $_POST['phone'] : '';
9 $email = isset($_POST['email']) ? $_POST['email'] : '';
10 $message = isset($_POST['message']) ? $_POST['message'] : '';
11
12 //Initialiser le tableau $erreurs
13 $erreurs = [];
14 //2.validator les données
15 // valider le nom
16 if(preg_match('/^([A-Za-z]{2,})+$/', $nom) === 0){
17     //ajouter un message d'erreur dans le tableau $erreurs
18     $erreurs["nom"] = "le nom n'est pas valide";
19 }
20 //valider le prenom
21 if(preg_match('/^([A-Za-z]{2,})+$/', $prenom) === 0){
22     //ajouter un message d'erreur dans le tableau $erreurs
23     $erreurs["prenom"] = "le prénom n'est pas valide";
24 }
25 //valider le numero de tel
26 if(preg_match('/^((?:\+\d{1,3}\d{1,3})|(?:\+\d{1,3}\d{1,3}\d{1,3}))[1-9]\d{1,9}\d{8}$/', $phone) === 0){
27     //ajouter un message d'erreur dans le tableau $erreurs
28     $erreurs["phone"] = "le numero de téléphone n'est pas valide";
29 }
30 //valider l'email
31 if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
32     $erreurs["email"] = "L'email n'est pas valide.";
33 }
34 // Validation du message
35 if (empty($message)) {
36     echo "Le message ne peut pas être vide.";
37 }
38 //3.mette en place la protection contre l'attaque XSS (failles)
39 $nom = htmlspecialchars($nom);
40 $prenom = htmlspecialchars($prenom);
41 $phone = htmlspecialchars($phone);
42 $email = htmlspecialchars($email);
43 $message = htmlspecialchars($message);
44
45 //4.si echec de validation,
46 //rediriger vers la page de formulaire avec des messages d'erreur et les données saisies
47 if (count($erreurs) > 0) { //si le tableau $erreurs n'est pas vide
48     $_SESSION["compte-donnees"]["nom"] = $nom;
49     $_SESSION["compte-donnees"]["prenom"] = $prenom;
50     $_SESSION["compte-donnees"]["phone"] = $phone;
51     $_SESSION["compte-donnees"]["email"] = $email;
52     $_SESSION["compte-donnees"]["message"] = $message;
53     $_SESSION["compte-erreurs"] = $erreurs;
54     // header("location: compte.php");//redirection avec le code HTTP 302
55     echo '<h2> erreur </h2>';
56     echo '<a href="contact.php">Retour au formulaire de contact</a>';
57     exit;
58 }
***** */

```

Développer la partie Back-end d'une application web et web mobile

À présent, je passe à la récupération de ces informations sécurisées et à leur insertion soigneuse dans la base de données, assurant ainsi l'intégrité et la sécurité des données.

Récupération des Valeurs du Formulaire et Sécurité :

Je récupère les valeurs du formulaire grâce à une boucle **foreach** sur le tableau **\$_POST**. Chaque valeur est sécurisée contre les attaques XSS à l'aide de la fonction **htmlspecialchars**. Les données sont ensuite stockées dans un tableau associatif **\$params** avec des clés correspondant aux champs du formulaire.

Connexion à la Base de Données et Vérification de l'Existence :

J'établis une connexion sécurisée à la base de données à l'aide de l'extension PHP Data Objects (**PDO**). Ensuite, je prépare une requête **SQL** pour vérifier si une demande similaire existe déjà dans la base de données en se basant sur l'adresse email et le message. Si une correspondance est trouvée, un message spécifique est affiché, et l'utilisateur est invité à retourner à la page d'accueil.

Insertion des Nouvelles Données :

Si aucune correspondance n'est trouvée, j'insère les nouvelles données sécurisées dans la table "contact" de la base de données via une nouvelle requête **SQL**. La connexion à la base de données est ensuite fermée avec **unset(\$cnn)**; pour des raisons de sécurité.

Gestion des Erreurs :

En cas d'erreur lors de l'exécution des requêtes **SQL**, une **exception PDO** est capturée. Les détails de l'erreur sont stockés dans la session pour un suivi ultérieur, et des informations spécifiques, telles que le nom, prénom, téléphone, email et message, sont également enregistrées dans la session. Un message d'erreur est affiché à l'utilisateur, l'invitant à contacter le support technique. Le processus se termine avec l'affichage du message d'erreur.

Cette approche sécurisée garantit la fiabilité des interactions avec la base de données tout en assurant une expérience utilisateur transparente.

```

55     ,
56     //Récupération des valeurs saisies et application sécurité
57     foreach ($_POST as $key => $val) {
58         $params[':' . $key] = (isset($_POST[$key]) && !empty($_POST[$key])) ? htmlspecialchars($_POST[$key]) : null;
59     }
60
61     include_once('./bdd/constants.bdd.php');
62     try {
63         // Connexion à la BDD
64         $cnn = new PDO("mysql:host=$db_host;dbname=$db_name;charset=utf8", $db_user, $db_pass);
65         // Options
66         $cnn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
67         $cnn->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_ASSOC);
68         // Teste si le nom n'existe pas déjà
69         $sql = 'SELECT COUNT(*) AS nb FROM contact WHERE email=:email AND message=:message'; // paramètre anonyme
70         $qry = $cnn->prepare($sql); // prépare la requête
71         $qry->execute(array(':email' => $email, ':message' => $message));
72         $row = $qry->fetch();
73
74         if ($row['nb'] == 1) {
75             echo '<h2 class="reponse_mailexiste">une demande de contact est déjà en cours !';
76             echo '<a href="homepage.php">Retour à l\'accueil</a>';
77         } else {
78             $sql = 'INSERT INTO contact(nom, prenom, phone, email, message) VALUES(:nom, :prenom, :phone, :email , :message)';
79             $qry = $cnn->prepare($sql);
80             $qry->execute($params);
81             unset($cnn); // déconnexion
82
83             echo '<h2 class="reponse_mailexiste">votre demande de contact a bien été envoyé !';
84             echo '<a href="abonnes.php"> Retour à la page Profil</a>';
85
86         }
87     } catch (PDOException $ex) {
88         // $ex contient l'exception capturée
89         $errorMessage = $ex->getMessage();
90         $_SESSION["compte-erreur-sql"] = $errorMessage;
91         $_SESSION["compte-donnees"]["nom"] = $nom;
92         $_SESSION["compte-donnees"]["prenom"] = $prenom;
93         $_SESSION["compte-donnees"]["phone"] = $phone;
94         $_SESSION["compte-donnees"]["email"] = $email;
95         $_SESSION["compte-donnees"]["message"] = $message;
96
97         echo $ex->getMessage();
98         exit;
99     }
100 ?>
101 ?>
102
103
104
105
106
107

```



Demande de contact

Pour toutes questions ou demandes, veuillez nous contacter via ce formulaire.

Nous vous répondrons dans les plus brefs délais.

elhas

zaza

0766876689

zaza@gmail.com

Bonjour, je souhaiterais un renseignement

Envoyer

Copyright © 2023. Projet Zahra.

[Politique de Confidentialité](#)

| Server : 127.0.0.1 > Base de données : zine_taxi > Table : contact | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|----------------------------------|------------|------------|--------------|---------------|----------------------------------|--|---------------------|--|------------|-----|--------|-------|-------|---------|-----------|--------------------------|----------------------------------|---|------|------|---------------|----------------------------------|-----------------------|---------------------|--------------------------|----------------------------------|---|----------|-------|---------------|-----------------|-----------------------|---------------------|--------------------------|----------------------------------|---|---------|-------|---------------|-------------------|-----------------------|---------------------|--------------------------|----------------------------------|---|------|------|---------------|----------------|-----------------------|---------------------|--------------------------|----------------------------------|---|-------|-------|------------|---------------|-------|---------------------|--------------------------|----------------------------------|---|--------|--------|------------|--------------|-------|---------------------|--------------------------|----------------------------------|---|-----|-----|------------|-----------------|-----------|---------------------|--------------------------|----------------------------------|----|-------|------|------------|----------------|--|---------------------|
| Parcourir | | Structure | SQL | Rechercher | Insérer | Exporter | Importer | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | Priviléges | Opérations | Déclencheurs | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Affichage des lignes 0 - 7 (total de 8, traitement en 0,0005 seconde(s).) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SELECT * FROM `contact` | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <input type="checkbox"/> Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <input type="checkbox"/> Tout afficher Nombre de lignes : <select>25</select> Filtrer les lignes : <input type="text"/> Chercher dans cette table Trier par clé : <input type="text"/> Aucun(e) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <input type="checkbox"/> Options supplémentaires | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th></th> <th></th> <th>id_contact</th> <th>nom</th> <th>prenom</th> <th>phone</th> <th>email</th> <th>message</th> <th>envoyé_le</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td>[Éditer] [Copier] [Supprimer]</td> <td>1</td> <td>toto</td> <td>toto</td> <td>0033612457845</td> <td>f0c163f3f614eb1ed1ecde5c8c0f9d67</td> <td>hello comment ca va ?</td> <td>2023-11-13 14:24:29</td> </tr> <tr> <td><input type="checkbox"/></td> <td>[Éditer] [Copier] [Supprimer]</td> <td>2</td> <td>hasnaoui</td> <td>zahra</td> <td>0033612457845</td> <td>zahra@gmail.com</td> <td>hello comment ca va ?</td> <td>2023-11-13 14:24:29</td> </tr> <tr> <td><input type="checkbox"/></td> <td>[Éditer] [Copier] [Supprimer]</td> <td>3</td> <td>bakkaou</td> <td>bakka</td> <td>0033612457845</td> <td>bakkaou@gmail.com</td> <td>hello comment ca va ?</td> <td>2023-11-13 14:24:29</td> </tr> <tr> <td><input type="checkbox"/></td> <td>[Éditer] [Copier] [Supprimer]</td> <td>4</td> <td>baka</td> <td>baka</td> <td>0033612457845</td> <td>baka@gmail.com</td> <td>hello comment ca va ?</td> <td>2023-11-13 14:24:29</td> </tr> <tr> <td><input type="checkbox"/></td> <td>[Éditer] [Copier] [Supprimer]</td> <td>5</td> <td>hulia</td> <td>hulia</td> <td>0766457896</td> <td>hulia@live.fr</td> <td>hello</td> <td>2023-11-13 14:24:29</td> </tr> <tr> <td><input type="checkbox"/></td> <td>[Éditer] [Copier] [Supprimer]</td> <td>6</td> <td>pascal</td> <td>pascal</td> <td>0766876689</td> <td>Dm@yahoo.com</td> <td>?????</td> <td>2023-11-13 14:24:29</td> </tr> <tr> <td><input type="checkbox"/></td> <td>[Éditer] [Copier] [Supprimer]</td> <td>9</td> <td>ali</td> <td>ali</td> <td>0766666666</td> <td>ali@hotmail.com</td> <td>C'EST MOI</td> <td>2023-11-14 23:26:01</td> </tr> <tr> <td><input type="checkbox"/></td> <td>[Éditer] [Copier] [Supprimer]</td> <td>12</td> <td>elhas</td> <td>zaza</td> <td>0766876689</td> <td>zaza@gmail.com</td> <td>Bonjour, je souhaiterais un renseignement</td> <td>2023-11-18 19:20:25</td> </tr> </tbody> </table> | | | | | | | | | | id_contact | nom | prenom | phone | email | message | envoyé_le | <input type="checkbox"/> | [Éditer] [Copier] [Supprimer] | 1 | toto | toto | 0033612457845 | f0c163f3f614eb1ed1ecde5c8c0f9d67 | hello comment ca va ? | 2023-11-13 14:24:29 | <input type="checkbox"/> | [Éditer] [Copier] [Supprimer] | 2 | hasnaoui | zahra | 0033612457845 | zahra@gmail.com | hello comment ca va ? | 2023-11-13 14:24:29 | <input type="checkbox"/> | [Éditer] [Copier] [Supprimer] | 3 | bakkaou | bakka | 0033612457845 | bakkaou@gmail.com | hello comment ca va ? | 2023-11-13 14:24:29 | <input type="checkbox"/> | [Éditer] [Copier] [Supprimer] | 4 | baka | baka | 0033612457845 | baka@gmail.com | hello comment ca va ? | 2023-11-13 14:24:29 | <input type="checkbox"/> | [Éditer] [Copier] [Supprimer] | 5 | hulia | hulia | 0766457896 | hulia@live.fr | hello | 2023-11-13 14:24:29 | <input type="checkbox"/> | [Éditer] [Copier] [Supprimer] | 6 | pascal | pascal | 0766876689 | Dm@yahoo.com | ????? | 2023-11-13 14:24:29 | <input type="checkbox"/> | [Éditer] [Copier] [Supprimer] | 9 | ali | ali | 0766666666 | ali@hotmail.com | C'EST MOI | 2023-11-14 23:26:01 | <input type="checkbox"/> | [Éditer] [Copier] [Supprimer] | 12 | elhas | zaza | 0766876689 | zaza@gmail.com | Bonjour, je souhaiterais un renseignement | 2023-11-18 19:20:25 |
| | | id_contact | nom | prenom | phone | email | message | envoyé_le | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <input type="checkbox"/> | [Éditer] [Copier] [Supprimer] | 1 | toto | toto | 0033612457845 | f0c163f3f614eb1ed1ecde5c8c0f9d67 | hello comment ca va ? | 2023-11-13 14:24:29 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <input type="checkbox"/> | [Éditer] [Copier] [Supprimer] | 2 | hasnaoui | zahra | 0033612457845 | zahra@gmail.com | hello comment ca va ? | 2023-11-13 14:24:29 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <input type="checkbox"/> | [Éditer] [Copier] [Supprimer] | 3 | bakkaou | bakka | 0033612457845 | bakkaou@gmail.com | hello comment ca va ? | 2023-11-13 14:24:29 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <input type="checkbox"/> | [Éditer] [Copier] [Supprimer] | 4 | baka | baka | 0033612457845 | baka@gmail.com | hello comment ca va ? | 2023-11-13 14:24:29 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <input type="checkbox"/> | [Éditer] [Copier] [Supprimer] | 5 | hulia | hulia | 0766457896 | hulia@live.fr | hello | 2023-11-13 14:24:29 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <input type="checkbox"/> | [Éditer] [Copier] [Supprimer] | 6 | pascal | pascal | 0766876689 | Dm@yahoo.com | ????? | 2023-11-13 14:24:29 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <input type="checkbox"/> | [Éditer] [Copier] [Supprimer] | 9 | ali | ali | 0766666666 | ali@hotmail.com | C'EST MOI | 2023-11-14 23:26:01 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <input type="checkbox"/> | [Éditer] [Copier] [Supprimer] | 12 | elhas | zaza | 0766876689 | zaza@gmail.com | Bonjour, je souhaiterais un renseignement | 2023-11-18 19:20:25 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <input type="checkbox"/> Tout cocher Avec la sélection : [Éditer] [Copier] [Supprimer] [Exporter] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Présentation du jeu d'essai élaboré par le candidat de la fonctionnalité la plus représentative (données en entrée, données attendues, données obtenues)

Introduction

Dans tous les sites web interagissant avec une base de données, quatre opérations fondamentales sont cruciales : la Création, la Lecture, la Modification et la Suppression, connues ensemble sous l'acronyme CRUD. Mettant en œuvre cette approche universellement reconnue, j'ai pu gérer efficacement les interactions des utilisateurs avec les données de mon site.

Fonctionnalités Utilisateur et CRUD

Dans cette section, je vais détailler comment les opérations CRUD sont appliquées aux fonctionnalités de base disponibles pour un utilisateur sur mon site web. Cela comprend la création de compte, l'accès et la modification de ses informations personnelles, ainsi que l'option de suppression de compte.

CREATE - Inscription des Utilisateurs

La première étape pour un utilisateur de mon site est de créer un compte. L'utilisateur fournit son nom, prénom, email et mot de passe qui sont alors enregistrés dans notre base de données.

READ - Affichage des Informations de l'Utilisateur

Une fois que l'utilisateur est connecté, son nom et prénom sont affichés sur la page d'accueil. De plus, toutes ses réservations, demandes de contact et devis sont récupérés de la base de données et affichés.

UPDATE - Changement de Mot de Passe

Si un utilisateur oublie son mot de passe, il a la possibilité de le réinitialiser grâce à une fonctionnalité dédiée. Pour ce faire, l'utilisateur doit d'abord entrer son adresse email. Un code lui est alors envoyé à cette adresse email, et il doit le renseigner sur le site. Une fois le code validé, l'utilisateur a la possibilité de définir un nouveau mot de passe. Ce dernier est ensuite mis à jour dans la base de données.

DELETE - Suppression d'un Compte

Un utilisateur a également la possibilité de supprimer son compte. Lorsqu'il choisit de le faire, toutes ses informations sont supprimées de la base de données.

Conclusion

Les opérations CRUD sont fondamentales pour toute application web interagissant avec une base de données. Sur mon site web, ces opérations permettent aux utilisateurs de créer un compte, de voir et mettre à jour leurs informations et enfin, de supprimer leur compte si nécessaire. Cela améliore grandement l'expérience utilisateur en rendant le site plus interactif.

CREATE - Incription des Utilisateurs

Page d'inscription :

J'ai créé une page d'inscription en PHP avec un formulaire. Cette page comprend des champs pour le nom, le prénom, l'email, et le mot de passe, chacun étant soumis à des validations côté client et serveur. En cas d'erreur, des messages spécifiques sont affichés. Le formulaire est configuré pour être soumis à "traitement_inscrip.php" lorsqu'il est validé. Un lien est également inclus pour rediriger les utilisateurs vers la page de connexion. L'en-tête (header.php) et le pied de page (footer.php) sont également inclus pour maintenir la cohérence du site.

```
1 <?php
2 include ("header.php");
3 >
4
5
6
7 <main class="login_signin">
8
9 <section class="section-inscription">
10    <h2 class="titre-in">Inscription</h2>
11    <p class="besoin">Besoin de faire une réservation ?</p>
12    <p class="besoin">Inscrivez-vous pour pouvoir réserver votre taxi.</p>
13
14    <form class="form-in" action="traitement_inscrip.php" method="POST">
15        <!-- Champs de saisie pour le nom avec une validation côté client via un pattern et côté serveur (PHP) si un message d'erreur est retourné -->
16        <input type="text" name="nom" id="nom" pattern="/[A-Za-z]{2,}[^ ]+" placeholder="Votre nom" required class="input_inscrip "><?php if(isset($nomMsgErreur) && !empty($nomMsgErreur)) echo ' is-invalid'; ?>" aria-describedby="nonFeedback" >
17        <?php if(isset($nomMsgErreur)){ ?>
18            <!-- Affichage d'un message d'erreur si la validation du nom du côté serveur échoue -->
19            <div class="invalid-feedback" id="nonFeedback">
20                <?php echo $nomMsgErreur; ?>
21            </div> </ph> } >
22
23        <!-- Champs de saisie pour le prénom avec une validation côté client via un pattern et côté serveur (PHP) si un message d'erreur est retourné -->
24        <input type="text" name="prenom" id="prenom" pattern="/[A-Za-z]{2,}[^ ]+" placeholder="Votre prénom" required class="input_inscrip "><?php if(isset($prenomMsgErreur) && !empty($prenomMsgErreur)) echo ' is-invalid'; ?>" aria-describedby="prenomFeedback" >
25        <?php if(isset($prenomMsgErreur)){ ?>
26            <!-- Affichage d'un message d'erreur si la validation du prénom du côté serveur échoue -->
27            <div class="invalid-feedback" id="prenomFeedback">
28                <?php echo $prenomMsgErreur; ?>
29            </div> </ph> } >
30
31        <!-- Champs de saisie pour l'email avec une validation côté client et côté serveur (PHP) si un message d'erreur est retourné -->
32        <input type="email" name="email" id="email" placeholder="Votre email" required class="input_inscrip "><?php if(isset($emailMsgErreur) && !empty($emailMsgErreur)) echo ' is-invalid'; ?>" aria-describedby="emailFeedback" >
33        <?php if(isset($emailMsgErreur)){ ?>
34            <!-- Affichage d'un message d'erreur si la validation de l'email du côté serveur échoue -->
35            <div class="invalid-feedback" id="emailFeedback">
36                <?php echo $emailMsgErreur; ?>
37            </div> </ph> } >
38
39        <!-- Champs de saisie pour le mot de passe avec une validation côté client via un pattern et côté serveur (PHP) si un message d'erreur est retourné -->
40        <input type="password" name="pass" id="mdp" pattern="*[A-Za-z0-9_]{8,}" placeholder="Créer votre mot de passe" required class="input_inscrip "><?php if(isset($passMsgErreur) && !empty($passMsgErreur)) echo ' is-invalid'; ?>" aria-describedby="passFeedback" >
41        <?php if(isset($passMsgErreur)){ ?>
42            <!-- Affichage d'un message d'erreur si la validation du mot de passe du côté serveur échoue -->
43            <div class="invalid-feedback" id="passFeedback">
44                <?php echo $passMsgErreur; ?>
45            </div> </ph> } >
46
47        <input class="input_inscrip" type="password" name="pass2" id="mdp2" placeholder="Confirmer votre mot de passe" required >
48
49    <button class="button_in" type="submit">Inscrivez-vous</button>
50
51 </main>
```

La vérification des mots de passe en Javascript :

J'ai intégré un script JavaScript pour la vérification de la correspondance entre les mots de passe lors de l'inscription. Ce script sélectionne le formulaire, écoute l'événement de soumission, puis compare les valeurs des champs de mot de passe. En cas de non-correspondance, une alerte est affichée, et l'envoi du formulaire est bloqué pour éviter des soumissions incorrectes.

```
52 <script>
53     //Vérification si les mots de passe sont les mêmes lors de l'inscription
54
55 // Sélection le formulaire dans ton JavaScript
56 var form = document.querySelector('.form-in');
57
58 // Ajoute un event listener pour l'événement "submit"
59 form.addEventListener('submit', function(e) {
60
61     // Sélection les champs de mot de passe
62     var pass = document.querySelector('#mdp');
63     var pass2 = document.querySelector('#mdp2');
64
65     // Vérifie si les mots de passe sont les mêmes
66     if (pass.value !== pass2.value) {
67
68         // Si les mots de passe ne sont pas les mêmes, affiche une erreur et bloque l'envoi du formulaire
69         alert('Les mots de passe ne correspondent pas!');
70         e.preventDefault();
71     }
72 });
73 </script>
74
75 <?php
76 include ("footer.php");
77 ?>
```

Le traitement de ma page inscription avec ma base de données :

Je commence par initialiser une session et inclure un fichier contenant les constantes de la base de données. Ensuite, je récupère les données d'un formulaire (nom, prénom, email, mot de passe) soumises via la méthode POST.

Je réalise une série de validations sur ces données, vérifiant notamment la validité du nom, du prénom, de l'email, et du mot de passe. En cas d'erreurs de validation, je les stocke dans un tableau d'erreurs.

Les caractères spéciaux dans le nom, prénom et email sont échappés pour des raisons de sécurité.

Le mot de passe est ensuite haché en utilisant la fonction password_hash de PHP, renforçant ainsi la sécurité des informations sensibles.

```
traitement_inscrip.php
1  <?php
2  // Démarrage de la session PHP
3  session_start();
4
5  // Inclusion du fichier de constantes pour la base de données
6  include("./bdd/constantsbdd.php");
7
8  // Récupération des données du formulaire (utilisation de l'opérateur ternaire pour éviter les erreurs)
9  $nom = isset($_POST['nom']) ? $_POST['nom'] : '';
10 $prenom = isset($_POST['prenom']) ? $_POST['prenom'] : '';
11 $email = isset($_POST['email']) ? $_POST['email'] : '';
12 $pass = isset($_POST['pass']) ? $_POST['pass'] : '';
13
14 // Tableau pour stocker les erreurs de validation
15 $erreurs = [];
16
17 // Validation des données du formulaire
18 if (preg_match("/^A-Za-z]{2,}+$/", $nom) === 0) {
19     $erreurs["nom"] = "Le nom n'est pas valide";
20 }
21
22 if (preg_match("/^A-Za-z]{2,}+$/", $prenom) === 0) {
23     $erreurs["prenom"] = "Le prénom n'est pas valide";
24 }
25
26 if (empty($email)) {
27     $erreurs['email'] = "Veuillez entrer votre adresse e-mail.";
28 } elseif (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
29     $erreurs['email'] = "L'adresse e-mail n'est pas valide.";
30 }
31
32 if (preg_match("/^A-Za-z0-9_){8,}/", $pass) === 0) {
33     $erreurs["pass"] = "Le mot de passe n'est pas valide";
34 }
35
36 // Nettoyage et sécurisation des données
37 $nom = htmlspecialchars($nom);
38 $prenom = htmlspecialchars($prenom);
39 $email = htmlspecialchars($email);
40
41 // Hachage du mot de passe pour des raisons de sécurité
42 $pass = password_hash($pass, PASSWORD_DEFAULT);
43
44 // Si des erreurs sont présentes, les stocker dans la session et afficher un message
45 if (count($erreurs) > 0) {
46     $_SESSION["compte-donnees"]["nom"] = $nom;
47     $_SESSION["compte-donnees"]["prenom"] = $prenom;
48     $_SESSION["compte-donnees"]["email"] = $email;
49     $_SESSION["compte-erreurs"] = $erreurs;
50
51     echo '<h2 class="reponse_mailexiste">Vous n\'avez pas rempli tous les champs. </h2>';
52     echo '<a href="connexion.php">Retour à l\'accueil</a>';
53     exit;
54 }
```

Dans cette partie du code PHP, je gère les erreurs éventuelles survenues lors de la validation des données du formulaire d'inscription. Si des erreurs sont détectées, je les stocke dans la session utilisateur, puis j'affiche un message d'erreur spécifique informant l'utilisateur des champs non remplis correctement. Enfin, le script se termine avec une redirection vers la page d'accueil de connexion.

En cas d'absence d'erreurs, j'essaie d'établir une connexion à la base de données MySQL en utilisant PDO. Si la connexion réussit, je vérifie si l'adresse email fournie lors de l'inscription existe déjà dans la base de données. Si c'est le cas, un message d'erreur est affiché, et l'utilisateur est invité à retourner à la page de connexion.

Si l'adresse email n'existe pas encore, les données (nom, prénom, email, mot de passe) sont insérées dans la table "inscription" de la base de données. Ensuite, la connexion à la base de données est fermée, et l'utilisateur est redirigé vers la page de connexion avec l'adresse email nouvellement enregistrée.

En cas d'erreur lors de la connexion à la base de données, je capture l'exception PDOException, stocke le message d'erreur dans la session utilisateur, puis affiche le message d'erreur à l'utilisateur avant de terminer le script.

```
55
56     try {
57         // Connexion à la base de données via PDO
58         $cnn = new PDO("mysql:host=$db_host;dbname=$db_name;charset=utf8", $db_user, $db_pass);
59         $cnn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
60         $cnn->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_ASSOC);
61
62         // Vérification si l'email existe déjà dans la base de données
63         $sql = 'SELECT COUNT(*) AS nb FROM inscription WHERE email=?';
64         $qry = $cnn->prepare($sql);
65         $qry->execute(array($email));
66         $row = $qry->fetch();
67
68         if ($row['nb'] == 1) {
69             echo '<h2 class="reponse_mailexiste">Cet email existe déjà !';
70             echo '<a href="connexion.php">Retour à la connexion</a>';
71         } else {
72             // Insertion des données dans la base de données
73             $sql = 'INSERT INTO inscription (nom, prenom, email, pass) VALUES(:nom, :prenom, :email, :pass)';
74             $qry = $cnn->prepare($sql);
75             $qry->execute(array(
76                 ':nom' => $nom,
77                 ':prenom' => $prenom,
78                 ':email' => $email,
79                 ':pass' => $pass
80             ));
81             // Libération des ressources de la base de données
82             unset($cnn);
83
84             // Stockage de l'email dans la session et redirection vers la page de connexion
85             $_SESSION['email'] = $email;
86             header('location:connexion.php');
87         }
88     } catch (PDOException $ex) {
89         // En cas d'erreur lié à la base de données, afficher un message d'erreur
90         $errorMessage = $ex->getMessage();
91         $_SESSION["compte-erreur-sql"] = $errorMessage;
92         $_SESSION["compte-donnees"]["nom"] = $nom;
93         $_SESSION["compte-donnees"]["prenom"] = $prenom;
94         $_SESSION["compte-donnees"]["email"] = $email;
95         $_SESSION["compte-donnees"]["pass"] = $pass;
96
97         echo "Erreur : " . $errorMessage;
98         exit;
99     }
100 ?>
```

Aperçu :

The screenshot shows the Zine Taxi website's inscription page. At the top, there is a logo with a car icon and the text "ZINE TAXI" followed by "Transport avec chauffeur". Below the logo, there is a navigation bar with links for "Accueil", "Réservation", "Devis", "Contact", and "Profil". The main content area has a title "Inscription" and a sub-section "Besoin de faire une réservation ?". It includes a call-to-action "Inscrivez-vous pour pouvoir réserver votre taxi." Below this, there is a form with five input fields: "Votre nom", "Votre prénom", "Votre email", "Créer votre mot de passe", and "Confirmer votre mot de passe". A red "S'inscrire" button is located at the bottom of the form. At the very bottom of the page, there is a copyright notice: "Copyright © 2023. Projet Zahra."

READ - Affichage des Informations de l'Utilisateur

Suite à la finalisation de son inscription, l'utilisateur est invité à se connecter afin de bénéficier de toutes les fonctionnalités de la plateforme, notamment la réservation de services et l'accès à sa page de profil personnalisée. Cette étape constitue une passerelle essentielle pour explorer et tirer pleinement parti des services offerts.

Page connexion :

J'ai conçu une page de connexion en PHP comportant un formulaire avec des champs pour l'email et le mot de passe. Lors de la soumission du formulaire, les données sont envoyées à "traitement_co.php" pour traitement ultérieur. Des liens sont inclus pour la réinitialisation du mot de passe ("forgott.php") et la redirection vers la page d'inscription ("inscription.php"). L'apparence de la page, y compris le titre, les placeholders, et les styles, est gérée via des classes CSS définies dans le code ou incluses depuis d'autres fichiers, tels que "header.php" et "footer.php".

```
<?php
include ("header.php");
?>
<body class="body_co">

<main class="login_sign">
    <section class="section-co">
        <h2 class="titre-co">Connexion</h2>
        <p class="besoin1">Connectez-vous pour accéder à votre espace personnel</p>
        <!-- formulaire de connexion -->
        <form class="form-co" action="traitement_co.php" method="POST">
            <!-- EMAIL -->
            <input class="input_co" type="email" name="email" id="email" placeholder="Votre email" required>
            <!-- MOT DE PASSE -->
            <input class="input_co" type="mdp" name="pass" id="mdp" placeholder="Votre mot de passe" required>
            <br>
            <a class="mdp" href="forgott.php"> Mot de passe oublié ?</a>
            <a class="inscr" href="inscription.php">S'inscrire</a>
        </form>
    </section>
</main>

<?php
include ("footer.php");
?>
</body>
```

J'ai développé un script PHP « traitement_co.php » pour gérer le processus de connexion à une base de données. Lorsque l'utilisateur soumet son email et son mot de passe via un formulaire, les données sont récupérées et sécurisées. Ensuite, le mot de passe est crypté avant d'établir une connexion à la base de données.

Le script effectue une requête SQL pour vérifier si l'email existe dans la base de données et si le compte est activé. En cas de succès, il compare le mot de passe fourni avec celui stocké dans la base de données, utilisant la fonction `password_verify()` pour une vérification sécurisée.

Si les informations d'identification sont valides, le script stocke le nom et le prénom de l'utilisateur dans les variables de session, marque l'utilisateur comme connecté, et le redirige vers la page "abonnes.php". En cas d'erreur, des messages appropriés sont affichés pour informer l'utilisateur sur la nature du problème, que ce soit un mot de passe incorrect, un utilisateur inconnu, ou une erreur de connexion à la base de données.

```
<?php

// Regarde les données en POST
$email = ( isset( $_POST['email'] ) && !empty( $_POST['email'] ) ) ? htmlspecialchars( $_POST['email'] ) : null;
$pass = ( isset($_POST['pass']) && !empty($_POST['pass']) ) ? htmlspecialchars( $_POST['pass'] ) : null;

if ($pass) {
    // ? Cryptage
    $pass = $_POST['pass'];

    // ? Connexion à la base de données
    try {
        // Connexion
        include_once("./bdd/constants.login.php");
        $conn = new PDO('mysql:host=' . HOST . ';dbname=' . DATA . ';port=' . PORT . ';charset=utf8', USER, PASS);
        // Gestion des attributs de connexion: Exception et retour du select
        $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        $conn->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_ASSOC);

        $query = $conn->prepare('SELECT * FROM inscription WHERE email=? AND active=?');
        $query->execute(array($email, 1));
        // Si la ligne est trouvée
        if ($query->rowCount()===1) {
            $row = $query->fetch();

            // Vérifier le mot de passe
            $hashed_password_from_db = $row['pass'];
            echo "Mot de passe entré par l'utilisateur: " . $_POST['pass'] . "<br>";
            echo "Hachage du mot de passe de la DB : " . $hashed_password_from_db . "<br>";

            if (password_verify($_POST['pass'], $hashed_password_from_db)) {
                // Stocker le nom et le prénom dans les variables de session
                $_SESSION['nom'] = $row['nom'];
                $_SESSION['prenom'] = $row['prenom'];

                // Marquer l'utilisateur comme connecté
                $_SESSION['utilisateur_connecte'] = true;
                $_SESSION['email'] = $email;
                header('location:abonnes.php');
            } else {
                echo "Mot de passe incorrect";
            }
        } else {
            echo "Utilisateur inconnu";
        }
    } catch (PDOException $e) {
        echo 'Erreur de connexion à la base de données: ' . $e->getMessage();
    }
}
?>
```

Aperçu :

The screenshot shows the Zine Taxi website's login page. At the top, there is a logo for "ZINE TAXI" featuring a stylized car icon and the text "Transport avec chauffeur". To the right of the logo are navigation links: "Accueil", "Réservation", "Devis", "Contact", and a user profile icon labeled "Profil". Below the header is a large, light gray rectangular area containing the login form. The form has a title "Connexion" at the top center. Below it is a placeholder text "Connectez-vous pour accéder à votre espace personnel". The form consists of two input fields: "Votre email" and "Votre mot de passe", both with red placeholder text. A red "Connexion" button is positioned below the inputs. At the bottom of the form, there are two small links: "Mot de passe oublié ?" and "S'inscrire". At the very bottom of the page, there is a footer with the text "Copyright © 2023, Projet Zahra." and "Politique de Confidentialité".

Page abonnés sur laquelle les informations seront affichées une fois l'utilisateur connecté :

Affichage du message « Bienvenue Nom Prénom » de l'utilisateur :

Je commence par inclure le fichier d'en-tête (header.php), le fichier de fonctions (functions.php) contenant une fonction nommée getInitials, ainsi que le code de connexion à la base de données (constants.bdd.php). Ces inclusions sont nécessaires pour utiliser les fonctionnalités définies dans ces fichiers.

Ensuite, je vérifie si l'utilisateur est connecté en vérifiant la présence de la variable de session 'utilisateur_connecte' et si sa valeur est égale à true. Si l'utilisateur est connecté, j'affiche un message de bienvenue personnalisé en utilisant le nom et le prénom stockés dans les variables de session 'nom' et 'prenom'. Ce message est encadré par une structure HTML avec une classe de conteneur.

Si l'utilisateur n'est pas connecté, un message indiquant qu'il n'est pas connecté est affiché, et une redirection vers la page de connexion (connexion.php) est effectuée à l'aide de la fonction header(). Cela garantit que seuls les utilisateurs connectés ont accès à cette page, renforçant ainsi la sécurité de l'application.

```
projet_perso > abonnes.php
1  <?php
2  include("header.php");
3  // Inclure le fichier de fonctions (contenant getInitials)
4  include_once('functions.php');
5  // Inclure le code de connexion
6  include_once("./bdd/constants.bdd.php");
7  // session_start();
8
9  // Vérification si l'utilisateur est connecté
10 if (isset($_SESSION['utilisateur_connecte']) && $_SESSION['utilisateur_connecte'] === true) {
11     // Afficher le nom d'utilisateur dans un h1
12     echo '<div class="profil-container">
13     <h1> Bienvenue ' . $_SESSION['nom'] . ' ' . $_SESSION['prenom'] . ' </h1>
14 </div>';
15 } else {
16     echo "Vous n'êtes pas connecté !";
17     header("location: connexion.php");
18 }
```

Aperçu :

The screenshot shows the Zine Taxi website. At the top left is the logo 'ZINE TAXI' with the tagline 'Transport avec chauffeur'. To the right are navigation links: Accueil, Réservation, Devis, and Contact. A circular profile picture of a person is on the far right. Below the header, there's a large button labeled 'Réserver'. A red arrow originates from this button and points down towards the center of the page. At the bottom center, the text '“ BIENVENUE ELH ZAHRA ”' is displayed in quotes. The background features a light gray gradient.

Récupération des réservations médicales, professionnelles et personnelles de l'utilisateur connecté et les afficher sur la page Abonnés :

Je crée une section HTML avec la classe "abonnee", comprenant un titre "Mes Réservations" (titre-resa). Ensuite, je vérifie si l'utilisateur est connecté en vérifiant la présence de la variable de session 'utilisateur_connecte' et si sa valeur est égale à true.

Si l'utilisateur est connecté, je récupère les données de réservation associées à son adresse e-mail. Trois types de réservations sont gérés : médicales, professionnelles et personnelles. Les réservations médicales sont extraites de la table "reservation_medicale", les réservations professionnelles de la table "reservation_pro", et les réservations personnelles de la table "reservation_perso".

Ces réservations sont affichées en utilisant la fonction "displayReservations", qui semble être définie ailleurs dans le code ou dans un fichier inclus. Cette fonction personnalisée prend en charge l'affichage des réservations médicales, professionnelles et personnelles, chacune ayant un titre spécifique.

Si l'utilisateur n'est pas connecté, un message indiquant qu'il n'est pas connecté est affiché.

```
<?php
// Récupérer les données de réservation en fonction de l'email de l'utilisateur
if (isset($_SESSION['utilisateur_connecte']) && $_SESSION['utilisateur_connecte'] === true) {
    try {
        $email = $_SESSION['email'];

        // Récupérer les réservations médicales
        $sqlMedicale = 'SELECT * FROM reservation_medicale WHERE email=:email';
        $qryMedicale = $pdo->prepare($sqlMedicale);
        $qryMedicale->execute(array(':email' => $email));
        $reservationsMedicale = $qryMedicale->fetchAll();

        // Récupérer les pièces jointes à la réservation médicale
        $sqlMedicale = 'SELECT bon_de_transport, attestation_carte_vitale FROM reservation_medicale WHERE email=?';
        $qryMedicale = $pdo->prepare($sqlMedicale);
        $qryMedicale->execute(array($email));
        $row = $qryMedicale->fetch();

        // Vérifie si $row est un tableau avant de tenter d'accéder aux données
        if ($row) {
            $bon_de_transport_path = $row['bon_de_transport'];
            $attestation_carte_vitale_path = $row['attestation_carte_vitale'];
        } else {
            // Si $row n'est pas un tableau
            $bon_de_transport_path = null;
            $attestation_carte_vitale_path = null;
        }

        // Récupérer les réservations professionnelles
        $sqlPro = 'SELECT * FROM reservation_pro WHERE email=:email';
        $qryPro = $pdo->prepare($sqlPro);
        $qryPro->execute(array(':email' => $email));
        $reservationsPro = $qryPro->fetchAll();

        // Récupérer les réservations personnelles
        $sqlPerso = 'SELECT * FROM reservation_perso WHERE email=:email';
        $qryPerso = $pdo->prepare($sqlPerso);
        $qryPerso->execute(array(':email' => $email));
        $reservationsPerso = $qryPerso->fetchAll();

        // Afficher les données de réservation
        echo '<h2>Réservations médicales</h2>';
        displayReservations($reservationsMedicale, 'medicale');

        echo '<h2>Réservations professionnelles</h2>';
        displayReservations($reservationsPro, 'pro');

        echo '<h2>Réservations personnelles</h2>';
        displayReservations($reservationsPerso, 'perso');

    } catch (PDOException $err) {
        echo 'Erreur : ' . $err->getMessage();
    }
} else {
    echo "Vous n'êtes pas connecté !";
}
```

Afficher les données de réservation dans un tableau en fonction du type de réservation :

J'ai appelé la fonction nommée "displayReservations" qui se charge d'afficher les réservations dans un tableau. Cette fonction prend en paramètre un tableau de réservations ainsi que le type de réservation (médicale, professionnelle, personnelle). En fonction du type, l'en-tête du tableau est adapté pour inclure les colonnes appropriées.

Pour chaque réservation, la fonction affiche les détails tels que la date, l'heure, le lieu de départ et d'arrivée. En cas de réservations médicales, la fonction crée également des liens vers les bons de transport et les attestations de carte vitale, facilitant ainsi l'accès à ces documents.

Si aucune réservation n'est présente, un message indiquant l'absence de réservations est affiché. En résumé, cette fonction contribue à une mise en page claire et organisée des informations de réservation, offrant aux utilisateurs une vue détaillée de leurs réservations en fonction de leur type.

```
// Fonction pour afficher les réservations dans un tableau
function displayReservations($reservations, $type)
{
    if (empty($reservations)) {
        echo "<p>Aucune réservation trouvée.</p>";
        return;
    }
    echo '<table border="1">';
    // On adapte l'en-tête du tableau en fonction du type de réservation
    if ($type == 'medicale') {
        echo '<tr><th>Date</th><th>Heure</th><th>Départ</th><th>Arrivée</th><th>Bon de transport</th><th>Attestation de carte vitale</th></tr>';
    } else if ($type == 'pro') {
        echo '<tr><th>Date</th><th>Heure</th><th>Départ</th><th>Arrivée</th></tr>';
    } else {
        echo '<tr><th>Date</th><th>Heure</th><th>Départ</th><th>Arrivée</th></tr>';
    }
    foreach ($reservations as $reservation) {
        echo '<tr>';
        echo "<td>" . date('d/m/Y', strtotime($reservation['date'])) . "</td>"; // on affiche la date en format français
        echo "<td>" . substr($reservation['heure'], 0, 5) . "</td>"; // On affiche que l'heure, pas les secondes
        echo "<td>" . $reservation["depart"] . "</td>";
        echo "<td>" . $reservation["arrivee"] . "</td>";
        // Pour les réservations médicales, on affiche aussi les colonnes pour le bon de transport et l'attestation de carte vitale
        if ($type == 'medicale') {
            echo "<td><a href='".$reservation["bon_de_transport"].">Voir le bon de transport</a></td>";
            echo "<td><a href='".$reservation["attestation_carte_vitale"].">Voir l'attestation de carte vitale</a></td>";
        }
        echo '</tr>';
    }
    echo '</table>';
}
?>
```

Aperçu :

| MES RÉSERVATIONS | | | | | |
|-------------------------------|-------|--------------------------------|--------------------------------|--|--|
| RÉSERVATIONS MÉDICALES | | | | | |
| Date | Heure | Départ | Arrivée | Bon de transport | Attestation de carte vitale |
| 24/11/2023 | 18:15 | 12 rue de la ville 77000 Melun | 12 rue de la ville 77000 Melun | Voir le bon de transport | Voir l'attestation de carte vitale |
| 30/11/2023 | 17:22 | 12 rue de la ville 77000 Melun | 12 rue de la ville 77000 Melun | Voir le bon de transport | Voir l'attestation de carte vitale |
| 02/12/2023 | 08:32 | 12 rue de la ville 77000 Melun | 12 rue de la ville 77000 Melun | Voir le bon de transport | Voir l'attestation de carte vitale |
| 22/11/2023 | 21:25 | 12 rue de la ville 77000 Melun | 12 rue de la ville 77000 Melun | Voir le bon de transport | Voir l'attestation de carte vitale |
| RÉSERVATIONS PROFESSIONNELLES | | | | | |
| Date | Heure | Départ | Arrivée | | |
| 09/12/2023 | 20:36 | 12 rue de la ville 77000 Melun | | 12 rue de la ville 77000 Melun | |
| RÉSERVATIONS PERSONNELLES | | | | | |
| Date | Heure | Départ | Arrivée | | |
| 05/11/2023 | 18:03 | 12 rue de la ville 77000 Melun | | 12 rue de la ville 77000 Melun | |

Récupération des demandes de devis et de contacts de l'utilisateur connecté et les afficher sur la page Abonnés :

Dans la continuation du code, j'ai créé deux tableaux, "\$contact" et "\$devis", que j'ai initialisés comme des tableaux vides. Ces tableaux servent à stocker les données relatives aux demandes de contact et aux devis associés à l'utilisateur connecté.

En vérifiant d'abord si l'utilisateur est connecté, j'accède ensuite à la base de données pour récupérer les informations nécessaires. Pour les demandes de contact, j'ai utilisé une requête SQL pour sélectionner toutes les entrées de la table "contact" correspondant à l'email de l'utilisateur. De même, pour les devis, j'ai effectué une requête similaire sur la table "devis".

En cas de réussite de l'exécution des requêtes, les résultats sont stockés dans les tableaux "\$contact" et "\$devis". En revanche, si une erreur survient lors de l'exécution de ces requêtes, un message d'erreur est affiché.

```
<?php
    // Récupérer les données de contact et devis en fonction de l'email de l'utilisateur connecté
    if (isset($_SESSION['utilisateur_connecte']) && $_SESSION['utilisateur_connecte'] === true) {
        try {
            $email = $_SESSION['email'];

            // Tu prépares et exécutes ta requête pour récupérer les demandes de contact
            $sql = 'SELECT * FROM contact WHERE email = :email';
            $qry = $pdo->prepare($sql);
            $qry->execute([':email' => $email]);
            $contact = $qry->fetchAll();

            // Pareil pour les devis
            $sql = 'SELECT * FROM devis WHERE email = :email';
            $qry = $pdo->prepare($sql);
            $qry->execute([':email' => $email]);
            $devis = $qry->fetchAll();
        } catch (PDOException $ex) {
            echo "Erreur : " . $ex->getMessage();
        }
    }
}
```

Afficher les données de contact et de devis dans un tableau :

J'ai mis en place l'affichage des demandes de contact et des devis associés à l'utilisateur connecté. Tout d'abord, j'affiche les demandes de contact dans un tableau en précisant la date d'envoi et le message associé.

Pour rendre la date plus lisible, j'ai ajusté la locale en français, converti la date de la demande en timestamp, puis converti ce timestamp en une date au format français avec l'heure.

Ensuite, j'affiche les demandes de devis dans un autre tableau, mettant en évidence diverses informations telles que le type de demande, la date, l'heure, le type d'aller-retour, le lieu de départ et d'arrivée, ainsi que le message associé.

```
echo "<br><h2>Demandes de contact </h2>";
echo "<table>";
echo "<tr><th>Envoyé le </th><th>Message </th></tr>";
foreach ($contact as $contact) {
    echo "<tr>";
    // Ajuste la locale en français
    setlocale(LC_TIME, 'fr_FR.utf8', 'fra');

    // Convertit la date en string en timestamp
    $timestamp = strtotime($contact['envoyé_le']);

    // Convertit le timestamp en date au format français et heure
    $date = strftime('%A %d %B %Y', $timestamp);
    $heure = date('H:i', $timestamp);

    echo "<td>" . $date . " à " . $heure . "</td>";
    // echo "<td>" . $contact['envoyé_le'] . "</td>";
    echo "<td>" . $contact['message'] . "</td>";
    echo "</tr>";
}
echo "</table>";

echo "<br><h2>Demandes de devis </h2>";
echo "<table>";
echo "<tr><th>Type</th><th>Date</th><th>Heure</th><th>Type AR</th><th>Départ</th><th>Arrivée</th><th>Message</th></tr>";
foreach ($devis as $devis) {
    echo "<tr>";
    echo "<td>" . $devis['type'] . "</td>";
    echo "<td>" . date('d/m/Y', strtotime($devis['date'])) . "</td>"; // On convertit la date en format français
    echo "<td>" . substr($devis['heure'], 0, 5) . "</td>"; // On affiche que l'heure, pas les secondes
    echo "<td>" . $devis['typeAR'] . "</td>";
    echo "<td>" . $devis['depart'] . "</td>";
    echo "<td>" . $devis['arrivee'] . "</td>";
    echo "<td>" . $devis['message'] . "</td>";
    echo "</tr>";
}
echo "</table>";
?>
```

Aperçu :

DEMANDES DE CONTACT

| Envoyé le | Message |
|--------------------------------|-------------------------------------|
| lundi 13 novembre 2023 à 14:24 | Hello |
| lundi 13 novembre 2023 à 14:25 | UPDATE - Changement de Mot de Passe |

DEMANDES DE DEVIS

| Type | Date | Heure | Type AR | Départ | Arrivée | Message |
|---------|------------|-------|--------------|--------------------------------|--------------------------------|---------|
| medical | 10/12/2023 | 16:31 | aller_simple | 12 rue de la ville 77000 Melun | 12 rue de la ville 77000 Melun | hiiiiii |

Page Header, j'ai modifié le lien de redirection de Réservation afin de rediriger l'utilisateur lorsqu'il n'est pas connecté à la page inscription et lorsqu'il est connecté à la page réservation :

```
<div class="navbar-links">
    <?php
        // Démarrer la session
        session_start();
    ?>

    <ul>
        <li><a href="homepage.php">Accueil</a></li>
        <!-- <li><a href="inscription.php">Réservation</a></li> -->
        <?php
            // Vérifier si l'utilisateur est connecté
            if (isset($_SESSION['utilisateur_connecte']) && $_SESSION['utilisateur_connecte'] === true) {
                // Si l'utilisateur est connecté, affiche le lien vers la page de réservation
                echo '<li><a href="reservation.php">Réservation</a></li>';
            } else {
                // Si l'utilisateur n'est pas connecté, affiche le lien vers la page d'inscription
                echo '<li><a href="inscription.php">Réservation</a></li>';
            }
        ?>
```

J'ai également affiché les initiales de l'utilisateur connecté à la place du logo Profil :

```
<div class="navbar-links">
    <ul>
        <?php
            // Inclure le fichier de fonctions (contenant getInitials)
            include_once('functions.php');

            // Vérifier si l'utilisateur est connecté
            if (isset($_SESSION['utilisateur_connecte']) && $_SESSION['utilisateur_connecte'] === true) {
                // Récupérer le nom et le prénom de l'utilisateur depuis la session
                $nom = $_SESSION['nom'];
                $prenom = $_SESSION['prenom'];

                // Afficher les initiales générées à partir du nom et du prénom
                echo '<li class="initial"><a href="abonnes.php"><span style="display: flex; align-items: center; justify-content: center; width: 70px; height: 70px; border-radius: 50%; background-color: grey;">' . getInitials($nom, $prenom) . '</span></a></li>';
            } else {
                // Si l'utilisateur n'est pas connecté, afficher un lien vers la page d'abonnés
                echo '<li class="profil-li">
                    <a href="abonnes.php" class="connexion">
                        <i class="fa-solid fa-user"></i> Profil
                    </a>
                </li>';
            }
        ?>
    </ul>
</div>
</header>
```

Grace à ce fichier function.php

```
<?php

// Définition d'une fonction nommée getInitials qui prend deux paramètres : $nom et $prenom
function getInitials($nom, $prenom) {
    // Concaténation des premières lettres du nom et du prénom pour former les initiales
    $initials = $nom[0] . $prenom[0];

    // Conversion des initiales en majuscules
    return strtoupper($initials);
}

?>
```

Aperçu :

The screenshot shows a website for "ZINE TAXI" with the tagline "Transport avec chauffeur". The navigation menu includes "Accueil", "Réservation", "Devis", and "Contact". A red arrow points from the "EZ" button in the top right corner to the "EZ" button on the website's header.

ZINE TAXI
Transport avec chauffeur

Accueil Réservation Devis Contact

“ BIENVENUE ELH ZAHRA ”

Réserver

UPDATE - Changement de Mot de Passe

Mon fichier « **forgott.php** » gère le processus de réinitialisation du mot de passe utilisateur. J'entame la procédure en établissant une connexion à la base de données MySQL. Ensuite, j'ai défini différentes étapes, notamment "**entrer_email**", "**entrer_code**" et "**entrer_mdp**", pour orchestrer le flux de travail de réinitialisation. L'utilisateur commence par fournir son email, déclenchant la génération et l'envoi d'un code de validation. Lors de la saisie du code reçu, suivi du nouveau mot de passe, je m'assure de vérifier la validité du code et d'effectuer l'enregistrement sécurisé du mot de passe. Chaque étape comporte des vérifications spécifiques, notamment la validation de l'email et la gestion des erreurs. J'ai également inclus des fonctions dédiées pour l'envoi d'email et la validation de l'adresse email. Les formulaires HTML correspondant à chaque étape sont intégrés dans le script, agrémentés de styles CSS et de liens vers des fichiers externes.

```
<?php
// Démarrage de la session
session_start();
// Initialisation du tableau d'erreurs
$error = array();
// Inclusion du fichier mail.php
require "mail.php";
// Connexion à la base de données
try {
    $pdo = new PDO("mysql:host=localhost;dbname=zine_taxi", "root", "");
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    die("Impossible de se connecter : " . $e->getMessage());
}
// Initialisation du mode par défaut à "entrer_email"
$mode = "entrer_email";
if(isset($_GET['mode'])) {
    $mode = $_GET['mode'];
}
// Vérification si des données sont postées
if(count($_POST) > 0) {
    switch ($mode) {
        case 'entrer_email':
            // Récupération de l'email posté
            $email = $_POST['email'];
            // Validation de l'email
            if(!filter_var($email, FILTER_VALIDATE_EMAIL) || !valid_email($email)) {
                $error[] = "Entrez un email valide";
            } else {
                // Stockage de l'email en session et envoi du mail
                $_SESSION['forgott']['email'] = $email;
                envoi_email($email);
                header("Location: forgott.php?mode=entrer_code");
                die;
            }
            break;
        case 'entrer_code':
            // Récupération du code posté
            $code = $_POST['code'];
            // Vérification de la validité du code
            $result = code_correct($code);
            if($result == "Le code est correct") {
                $_SESSION['forgott']['code'] = $code;
                header("Location: forgott.php?mode=entrer_mdp");
                die;
            } else {
                $error[] = $result;
            }
            break;
        case 'entrer_mdp':
            // Récupération du mot de passe posté
            $password = $_POST['pass'];
            $password2 = $_POST['pass2'];
            // Vérification de la correspondance des mots de passe
            if($password !== $password2) {
                $error[] = "Les mots de passe ne correspondent pas";
            } elseif(!isset($_SESSION['forgott']['email']) || !isset($_SESSION['forgott']['code'])) {
                header("Location: forgott.php");
                die;
            } else {
                // Enregistrement du nouveau mot de passe et redirection
                enregistre_mdp($password);
                if(isset($_SESSION['forgott'])) {
                    unset($_SESSION['forgott']);
                }
                header("Location: connexion.php");
                die;
            }
            break;
        default:
            // Code par défaut
            break;
    }
}
}
```

Fonction pour l'envoi de l'email ainsi que le contenu de l'email :

```
// Fonction pour l'envoi de l'email
function envoi_email($email) {
    global $pdo;

    // Génération d'un code et d'une date d'expiration
    $expire = time() + (60 * 2);
    $code = rand(100000, 999999);
    $email = addslashes($email);

    // Insertion du code et de l'expiration dans la base de données
    $query = "INSERT INTO codes (email, code, expire) VALUES (?, ?, ?)";
    $stmt = $pdo->prepare($query);
    $stmt->execute([$email, $code, $expire]);

    // Envoi du mail avec le code généré

    send_mail($email, '[ZINE TAXI] Réinitialisation mot de passe', '
<head>
    meta charset="UTF-8"
    meta http-equiv="X-UA-Compatible" content="IE-edge"
    meta name="viewport" content="width=device-width, initial-scale=1.0"
    link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" integrity="sha512-incl1mask17CvqkXNQ/ZH/XLiwZO3yj7YtccnepD1ypA5ozpmT/E0IPmFB46ZmdtAc9eNBvH0H/ZpIBw==</link>
    link rel="stylesheet" href="email.css"
<style>
body {
    font-family: PT Sans, sans-serif;
    color: #333;
    background-color: #f6f6f6;
    margin: 0;
    padding: 0;
}
.box {
    padding: 20px;
    margin: auto;
    width: 98%;
    max-width: 800px;
    background-color: #fff;
    box-shadow: 0 0 10px rgba(0,0,0,0.1);
}
.logo {
    width: 200px;
    display: block;
    margin: 0 auto 30px;
}
.titre_email {
    text-align: center;
}
p {
    font-size: 16px;
    line-height: 1.5;
    margin-bottom: 20px;
}
</style>
</head>
<body>
    <div class="box">
        
        <h1 class="titre_email">ZINE TAXI</h1>
        <p>Coucou! Apparemment tu as oublié ton mot de passe, ne t'inquiète pas, nous allons te réinitialiser grâce à ce code : <b>' . $code . '</b></p>
        <p>Il va falloir retenir ce code et le copier sur le site afin de changer le mot de passe de ton compte.</p>
    </div>
</body>
</html>
');
}
```

Fonctions pour l'enregistrement du nouveau mot de passe et la validation de l'email :

```
// Fonction pour l'enregistrement du nouveau mot de passe
function enregistre_mdp($password) {
    global $pdo;
    $password = sha1(md5($password) . md5($password));
    $email = addslashes($_SESSION['forgott']['email']);

    // Mise à jour du mot de passe dans la base de données
    $query = "UPDATE inscription SET pass = ? WHERE email = ? LIMIT 1";
    $stmt = $pdo->prepare($query);
    $stmt->execute([$password, $email]);
}

// Fonction pour la validation de l'adresse email
function valid_email($email) {
    global $pdo;
    $email = addslashes($email);

    // Recherche de l'adresse email dans la base de données
    $query = "SELECT * FROM inscription WHERE email = ? LIMIT 1";
    $stmt = $pdo->prepare($query);
    $stmt->execute([$email]);
    $result = $stmt->fetch(PDO::FETCH_ASSOC);

    return $result ? true : false;
}
```

Les différentes fonctions pour l'enregistrement du mot de passe, pour la validation de l'adresse email et pour la vérification du code :

```
// Fonction pour l'enregistrement du nouveau mot de passe
function enregistre_mdp($password) {
    global $pdo;
    $password = sha1(md5($password) . md5($password));
    $email = addslashes($_SESSION['forgott']['email']);

    // Mise à jour du mot de passe dans la base de données
    $query = "UPDATE inscription SET pass = ? WHERE email = ? LIMIT 1";
    $stmt = $pdo->prepare($query);
    $stmt->execute([$password, $email]);
}

// Fonction pour la validation de l'adresse email
function valid_email($email) {
    global $pdo;
    $email = addslashes($email);

    // Recherche de l'adresse email dans la base de données
    $query = "SELECT * FROM inscription WHERE email = ? LIMIT 1";
    $stmt = $pdo->prepare($query);
    $stmt->execute([$email]);
    $result = $stmt->fetch(PDO::FETCH_ASSOC);

    return $result ? true : false;
}

// Fonction pour la vérification du code
function code_correct($code) {
    global $pdo;

    $code = addslashes($code);
    $expire = time();
    $email = addslashes($_SESSION['forgott']['email']);
    // Recherche du code dans la base de données
    $query = "SELECT * FROM codes WHERE code = ? AND email = ? ORDER BY id DESC LIMIT 1";
    $stmt = $pdo->prepare($query);
    $stmt->execute([$code, $email]);
    $result = $stmt->fetch(PDO::FETCH_ASSOC);

    if($result) {
        if($result['expire'] > $expire) {
            return "Le code est correct";
        } else {
            return "Le code a expiré";
        }
    } else {
        return "Le code est incorrect";
    }
}
```

Les formulaires HTML sont structurés de manière conditionnelle en utilisant une construction basée sur le mode grâce à une instruction « switch-case » :

```
<!-- Début du document HTML -->
<!DOCTYPE html>
<html lang="fr">
<head>
    <!-- Balises d'en-tête du document -->
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>Mot de passe oublié</title>
</head>

<?php
// Inclusion du fichier header.php
include("header.php");
?>

<body class="body_mdp">
<!-- Corps du document HTML -->
<div class="container">
    <?php
        // Structure conditionnelle basée sur le mode
        switch ($mode) {
            case 'entrer_email':
                // Formulaire pour entrer l'email
            ?>
                <main class="main_mdp">
                    <div class="cadre">
                        <div class="cadre2">
                            <form class="form_mdp" method="post" action="forgott.php?mode=entrer_email">
                                <h1 class="titre_mdp">Mot de passe oublié ?</h1>
                                <?php
                                    foreach ($error as $err) {
                                        echo $err . "<br>";
                                    }
                                ?>
                                <label for="email">Entrez votre email</label><br>
                                <input class="input_mdp" type="email" name="email" placeholder="Votre email"><br>
                                <br style="clear: both;">
                                <input class="input-mdp" type="submit" value="Suivant">
                                <br><br>
                                <div>
                                    <a class="link_mdp" href="connexion.php">Connexion</a>
                                </div>
                            </form>
                        </div>
                    </main>
                <?php
                    // Inclusion du fichier footer.php
                    include("footer.php");
                    break;
                
```

```

case 'entrer_code':
// Formulaire pour entrer le code
?>
<main class="main_mdp">
    <div class="cadre">
        <div class="cadre2">
            <form method="post" action="forgott.php?mode=entrer_code">
                <h1 class="titre_mdp">Mot de passe oublié ?</h1>
                <h3>Entrez le code envoyé à : <?php echo $_SESSION['forgott']['email']; ?></h3>
                <?php
                    foreach ($error as $err) {
                        echo $err . "<br>";
                    }
                ?>
                <input class="input_mdp" type="password" name="code" placeholder="12345"><br>
                <br style="clear: both;">
                <a href="forgott.php">
                    <input class="input-mdp" type="button" value="Retour">
                </a>
                <input class="input-mdp" type="submit" value="Suivant">
                <br><br>
                <div>
                    <a class="link_mdp" href="connexion.php">Connexion</a>
                </div>
            </form>
        </div>
    </main>
<?php
// Inclusion du fichier footer.php
include("footer.php");
break;

```

```

case 'entrer_mdp':
// Formulaire pour entrer le nouveau mot de passe
?>
<main class="main_mdp">
    <div class="cadre">
        <div class="cadre2">
            <form method="post" action="forgott.php?mode=entrer_mdp">
                <h1 class="titre_mdp">Mot de passe oublié</h1>
                <h3>Entrez le nouveau mot de passe</h3>
                <?php
                    foreach ($error as $err) {
                        echo $err . "<br>";
                    }
                ?>
                <input class="input_mdp" type="password" name="pass" placeholder="Nouveau mot de passe"><br>
                <input class="input_mdp" type="password" name="pass2" placeholder="Nouveau mot de passe"><br>
                <br style="clear: both;">
                <a href="forgott.php">
                    <input class="input-mdp" type="button" value="Retour">
                </a>
                <input class="input-mdp" type="submit" value="Suivant">
                <br><br>
                <div>
                    <a class="link_mdp" href="connexion.php">Connexion</a>
                </div>
            </form>
        </div>
    </main>
<?php
// Inclusion du fichier footer.php
include("footer.php");
break;

default:
// Code par défaut
break;
?>
</div>
</body>
</html>

```

Aperçu :

The screenshot shows a password recovery form titled "Mot de passe oublié ?". It asks for an email address and has "Suivant" and "Connexion" buttons.

ZINE TAXI
Transport avec chauffeur

Accueil Réservation Devis Contact Profil

Mot de passe oublié ?
Entrez votre email
Votre email
Suivant
Connexion

Copyright © 2023. Projet Zahra.

The screenshot shows a step where a code is entered. It displays "Entrez le code envoyé à : zahra.elhasnaoui92@gmail.com" and a field containing "12345". It has "Retour", "Suivant", and "Connexion" buttons.

ZINE TAXI
Transport avec chauffeur

Accueil Réservation Devis Contact Profil

Mot de passe oublié ?
Entrez le code envoyé à :
zahra.elhasnaoui92@gmail.com
12345
Retour
Suivant
Connexion

The screenshot shows a step where a new password is created. It has fields for "Nouveau mot de passe" and "Confirmation mot de passe", both containing "Nouveau mot de passe". It has "Retour", "Suivant", and "Connexion" buttons.

ZINE TAXI
Transport avec chauffeur

Accueil Réservation Devis Contact Profil

Mot de passe oublié
Entrez le nouveau mot de passe
Nouveau mot de passe
Confirmation mot de passe
Retour
Suivant
Connexion

Copyright © 2023. Projet Zahra.

DELETE - Suppression d'un Compte

Pour gérer la suppression définitive d'un compte utilisateur après confirmation via un formulaire dédié. Je commence par vérifier si un utilisateur est actuellement connecté. En cas de soumission du formulaire de suppression, j'établis une connexion à la base de données, récupère l'identifiant de l'utilisateur, supprime ses données dans différentes tables, le déconnecte et le redirige vers la page d'accueil avec un message de confirmation. Des messages d'erreur sont également générés pour assurer un fonctionnement fluide.

```
<?php
// Inclusion du fichier de configuration des constantes
include_once("./bdd/constants.login.php");

// Démarrage de la session
session_start();

// Vérification de l'existence d'un utilisateur connecté en session
if (isset($_SESSION['utilisateur_connecte']) === true) {
    // Vérification de la soumission du formulaire de suppression
    if (isset($_POST['confirmer_suppression'])) {
        try {
            // Connexion à la base de données
            $cnn = new PDO('mysql:host=' . HOST . ';dbname=' . DATA . ';port=' . PORT . ';charset=utf8', USER, PASS);
            $cnn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

            // Récupération de l'email de l'utilisateur connecté
            $email = $_SESSION['email'];

            // Requête pour obtenir l'id_inscription associé à l'email
            $sql = 'SELECT id_inscription FROM inscription WHERE email = :email';
            $qry = $cnn->prepare($sql);
            $qry->execute(array(':email' => $email));
            $result = $qry->fetch();

            // Initialisation de l'id_inscription
            $idInscription = null;

            // Vérification de la réussite de la requête
            if ($result !== false) {
                // Récupération de l'id_inscription
                $idInscription = $result['id_inscription'];

                // Tableaux des tables à vider
                $tables = ['reservation_medicale', 'reservation_pro', 'reservation_perso', 'contact', 'devis', 'inscription'];

                // Suppression des données de chaque table pour l'utilisateur
                foreach ($tables as $table) {
                    // Détermination du paramètre et de la valeur selon la table
                    $param = ($table == 'inscription') ? ':id_inscription' : ':email';
                    $value = ($table == 'inscription') ? $idInscription : $email;

                    // Requête de suppression pour la table
                    $sql = "DELETE FROM $table WHERE " . ltrim($param, ':') . " = :value";
                    $qryDelete = $cnn->prepare($sql);
                    $qryDelete->execute(array(':value' => $value));
                }
            }

            // Déconnexion de l'utilisateur
            session_destroy();

            // Redirection après suppression avec un message
            header("Location: homepage.php?message=Votre%20compte%20a%20bien%20été%20supprimé");
            exit();
        } else {
            echo 'Aucun résultat trouvé pour cette requête.';
        }
    }

    // Fermeture de la connexion
    unset($cnn);
} catch (PDOException $err) {
    // Gestion des erreurs de la base de données
    echo 'Erreur : ' . $err->getMessage();
}
} else {
    // Message si le formulaire n'a pas été soumis
    echo 'Le formulaire n\'a pas été soumis.';
}
} else {
    // Message si l'utilisateur n'est pas connecté
    echo "Vous n'êtes pas connecté !";
}
?>
```

Le formulaire dédié :

```
<?php include("header.php"); ?>
<main>
    <div class="cadre">
        <h1>Suppression de compte</h1>
        <form method="POST" action="">
            <p>Es-tu sûr de vouloir supprimer ton compte ? Cette action est irréversible !</p>
            <button type="submit" name="confirmer_suppression">Oui, je suis sûr</button>
        </form>
    </div>
</main>
<?php include("footer.php"); ?>
```

Aperçu :

The screenshot shows the Zine Taxi website with a navigation bar at the top featuring the logo, menu items (Accueil, Réservation, Devis, Contact), and a user icon labeled 'EZ'. The main content area has a dark background with white text. It displays the heading 'Suppression de compte' and a message asking if the user is sure they want to delete their account. A red button labeled 'Oui, je suis sûr' (Yes, I'm sure) is centered below the message.

Après confirmation, l'utilisateur sera automatiquement déconnecté et dirigé vers la page d'accueil où un message lui indiquera que son compte a été supprimé avec succès.

The screenshot shows the Zine Taxi website with a navigation bar at the top featuring the logo, menu items (Accueil, Réservation, Devis, Contact), and a user icon labeled 'Profil'. The main content area has a light background with pink text. It displays a success message 'Votre compte a bien été supprimé' (Your account has been successfully deleted). Below this, there's a section titled 'Besoin d'un taxi ?' (Need a taxi?) with sub-sections 'Un taxi pour tous vos déplacements' (A taxi for all your trips) and 'Faites-nous confiance pour vous conduire ou venir vous chercher aux aéroports et aux gares.' (Trust us to drive you or pick you up from airports and stations.). There are also images of people in a car and a woman with luggage.

Description de la veille, effectuée par le candidat durant le projet, sur les vulnérabilités de sécurité

La mise en place de mesures de sécurité constitue une étape essentielle dans le développement d'un site internet. À cette fin, j'ai intégré plusieurs fonctionnalités sécuritaires afin d'assurer une protection adéquate :

htmlspecialchars : Lorsque les utilisateurs saisissent des informations, j'utilise htmlspecialchars pour prévenir les attaques XSS. Cette fonction transforme les caractères spéciaux en leur équivalent HTML, assurant un affichage normal tout en évitant toute injection de code malveillant.

La fonction filter_var() : Pour filtrer les données saisies, j'emploie la fonction filter_var(). Par exemple, je vérifie que les adresses e-mail sont au format valide, renforçant ainsi la qualité des informations reçues.

La fonction password_hash() : Pour sécuriser les mots de passe stockés, j'utilise la fonction password_hash(). Cette dernière crypte les mots de passe, rendant leur lecture impossible même en cas d'accès non autorisé à la base de données.

la fonction password_verify : Pour comparer le mot de passe fourni lors de la tentative de connexion avec le mot de passe haché stocké dans la base de données. L'utilisation de cette fonction garantit une comparaison sécurisée.

Utilisation des sessions : Le démarrage de sessions avec session_start() et l'utilisation de \$_SESSION pour vérifier l'existence d'un utilisateur connecté garantissent l'accès sécurisé aux fonctionnalités sensibles du site.

Requêtes préparées : Les requêtes SQL sont préparées (utilisation de \$cnn->prepare()), minimisant ainsi les risques d'injections SQL et renforçant la sécurité de l'accès à la base de données.

Nettoyage des données : L'utilisation de ltrim() pour éliminer les caractères indésirables contribue à assurer l'intégrité des données tout au long du processus, évitant toute manipulation malveillante.

Utilisation de unset pour fermer la connexion : La fermeture sécurisée de la connexion à la base de données est assurée grâce à unset(\$cnn), contribuant à une gestion appropriée des ressources.

Protection contre les attaques CSRF : Bien que non explicitement dans le code fourni, l'ajout de jetons anti-CSRF dans les formulaires renforcerait la protection contre les attaques de type CSRF.

Messages génériques en cas d'échec : En cas d'échec de la requête, des messages génériques sont affichés, évitant la divulgation d'informations sensibles tout en informant l'utilisateur de manière appropriée.

Gestion sécurisée des fichiers uploadés : Si des fichiers sont uploadés, je m'assure qu'ils sont stockés de manière sécurisée et que les liens vers ces fichiers ne peuvent pas être exploités à des fins malveillantes.

En somme, ces fonctionnalités combinées assurent une approche robuste de la sécurité, protégeant ainsi les données des utilisateurs à toutes les étapes du processus.

Description d'une situation de travail ayant nécessité une recherche, effectuée par le candidat durant le projet, à partir de site anglophone

strftime — Formate une date locale avec la configuration locale :

Dans le cadre de mon projet, j'ai choisi de formater l'affichage de la date en utilisant la fonction **strftime** pour obtenir un format français. Cette approche permet une présentation plus conviviale et adaptée à l'utilisateur, offrant une meilleure expérience en affichant la date dans un format localisé.

J'ai alors effectué plusieurs recherches et j'ai trouvé l'explication sur ce site :

<https://www.php.net/manual/en/function.strftime.php>

Voici un extrait en anglais :

The screenshot shows a screenshot of the PHP documentation website. At the top, there is a blue header bar with the 'php' logo, 'Downloads', 'Documentation', 'Get Involved', 'Help', and 'php 8.2'. Below the header, a dark navigation bar contains links for 'PHP Manual', 'Function Reference', 'Date and Time Related Extensions', 'Date/Time', and 'Date/Time Functions'. On the right side of the header, there is a 'Change language:' dropdown set to 'English', and links for 'Submit a Pull Request' and 'Report a Bug'. The main content area has a light gray background. The title 'strftime' is displayed in a large, bold, dark purple font. Below the title, the text '(PHP 4, PHP 5, PHP 7, PHP 8)' is shown in parentheses. A brief description follows: 'strftime — Format a local time/date according to locale settings'. Under the heading 'Description', the function signature 'strftime(string \$format, ?int \$timestamp = null): string|false' is provided in a code block. A detailed description below states: 'Format the time and/or date according to locale settings. Month and weekday names and other language-dependent strings respect the current locale set with [setlocale\(\)](#)'. The bottom of the page features a footer with the text '2023 | EL HASNAOUI ZAHRA | TITRE PROFESSIONNEL DÉVELOPPEUR WEB ET WEB MOBILE' and a red '63' in the bottom right corner.

The following characters are recognized in the `format` parameter string

| format | Description | Example returned values |
|---------------|--|---|
| <i>Day</i> | --- | --- |
| %a | An abbreviated textual representation of the day | Sun through Sat |
| %A | A full textual representation of the day | Sunday through Saturday |
| %d | Two-digit day of the month (with leading zeros) | 01 to 31 |
| %e | Day of the month, with a space preceding single digits. Not implemented as described on Windows. See below for more information. | 1 to 31 |
| %j | Day of the year, 3 digits with leading zeros | 001 to 366 |
| %u | ISO-8601 numeric representation of the day of the week | 1 (for Monday) through 7 (for Sunday) |
| %w | Numeric representation of the day of the week | 0 (for Sunday) through 6 (for Saturday) |
| <i>Month</i> | --- | --- |
| %b | Abbreviated month name, based on the locale | Jan through Dec |
| %B | Full month name, based on the locale | January through December |
| %h | Abbreviated month name, based on the locale (an alias of %b) | Jan through Dec |
| %m | Two digit representation of the month | 01 (for January) through 12 (for December) |
| <i>Year</i> | --- | --- |
| %C | Two digit representation of the century (year divided by 100, truncated to an integer) | 19 for the 20th Century |
| %g | Two digit representation of the year going by ISO-8601:1988 standards (see %V) | Example: 09 for the week of January 6, 2009 |
| %G | The full four-digit version of %g | Example: 2008 for the week of January 3, 2009 |
| %y | Two digit representation of the year | Example: 09 for 2009, 79 for 1979 |
| %Y | Four digit representation for the year | Example: 2038 |

This line of code, `setlocale(LC_TIME, 'fr_FR.utf8','fra');`, added to my PHP script, aims to adjust the locale setting specifically for time management. In particular, I have set the locale to French using the `setlocale` function.

The `setlocale` function takes two main arguments:

The first one, `LC_TIME`, indicates that I want to adjust the locale for time-related formats such as dates. The second argument, `'fr_FR.utf8','fra'`, specifies the exact locale. By specifying `'fr_FR.utf8'`, I ensure that the UTF-8 format is used for French, and `'fra'` is a shorter alternative for the same locale.

By adjusting the locale in this way, I ensure that all time-related operations, such as date display, follow French conventions. This contributes to making my application more user-friendly and suitable for a French-speaking audience.

```
// Ajuste le locale en français  
setlocale(LC_TIME, 'fr_FR.utf8','fra');
```

After extracting the date of the message, which is stored as a timestamp in the variable `$contact['envoyé_le']`, I use the `strtotime` function to convert this timestamp into a numerical representation of the moment when the message was sent.

```
// Convertit la date en string en timestamp  
$timestamp = strtotime($contact['envoyé_le']);
```

Then, to make this information more readable, I use the `strftime` function to convert it into a string. This allows me to format the date according to my preferences, specifically in the French format with the full day of the week, the day of the month in numeric format, the full month name, and the year in a 4-digit format. This way, my users can easily understand when the message was sent.

```
// Convertit le timestamp en date au format français et heure  
// %A(Nom complet du jour de la semaine) / %d(Jour du mois en numérique, sur 2 chiffres (avec le zéro initial)) / %B(Nom complet du mois, suivant la locale) / %Y(L'année, sur 4 chiffres)  
$date = strftime('%A %d %B %Y', $timestamp);
```

Overview :

DEMANDES DE CONTACT

| Envoyé le | Message |
|--------------------------------|-------------------------------------|
| lundi 13 novembre 2023 à 14:24 | Hello |
| lundi 13 novembre 2023 à 14:25 | UPDATE - Changement de Mot de Passe |

En français :

`strftime` — Formate une date/heure locale avec la configuration locale.

`strftime(string $format, ?int $timestamp = null): string|false`

Formate une date et/ou une heure suivant la localisation locale. Les noms des mois, des jours de la semaine mais aussi d'autres chaînes dépendant de la localisation, respecteront la localisation courante définie par la fonction `setlocale()`.

Les caractères suivants sont reconnus dans le paramètre `format`

| format | Description | Exemple de valeurs renvoyées |
|---------------|---|--|
| <i>Jour</i> | --- | --- |
| %a | Nom abrégé du jour de la semaine | De Sun à Sat |
| %A | Nom complet du jour de la semaine | De Sunday à Saturday |
| %d | Jour du mois en numérique, sur 2 chiffres (avec le zéro initial) | De 01 à 31 |
| %e | Jour du mois, avec un espace précédant le premier chiffre. L'implémentation Windows est différente, voyez après pour plus d'informations. | De 1 à 31 |
| %j | Jour de l'année, sur 3 chiffres avec un zéro initial | 001 à 366 |
| %u | Représentation ISO-8601 du jour de la semaine | De 1 (pour Lundi) à 7 (pour Dimanche) |
| %w | Représentation numérique du jour de la semaine | De 0 (pour Dimanche) à 6 (pour Samedi) |
| <i>Mois</i> | | |
| %b | Nom du mois, abrégé, suivant la locale | De Jan à Dec |
| %B | Nom complet du mois, suivant la locale | De January à December |
| <i>Année</i> | | |
| %C | Représentation, sur 2 chiffres, du siècle (année divisée par 100, réduit à un entier) | 19 pour le 20ème siècle |
| %g | Représentation, sur 2 chiffres, de l'année, compatible avec les standards ISO-8601:1988 (voyez %V) | Exemple : 09 pour la semaine du 6 janvier 2009 |
| %G | La version complète à quatre chiffres de %g | Exemple : 2009 pour la semaine du 3 janvier 2009 |
| %y | L'année, sur 2 chiffres | Exemple : 09 pour 2009, 79 pour 1979 |
| %Y | L'année, sur 4 chiffres | Exemple : 2038 |

Cette ligne de code, `setlocale(LC_TIME, 'fr_FR.utf8','fra');`, que j'ai ajoutée à mon script PHP, a pour objectif d'ajuster le paramètre de localisation (locale) spécifique à la gestion du temps. En particulier, j'ai défini la localisation en français à l'aide de la fonction `setlocale`.

La fonction `setlocale` prend deux arguments principaux :

Le premier, `LC_TIME`, indique que je souhaite ajuster la localisation pour les formats de temps tels que les dates.

Le deuxième argument, `'fr_FR.utf8','fra'`, précise la localisation exacte. En spécifiant `'fr_FR.utf8'`, je m'assure que le format **UTF-8** est utilisé pour le français, et `'fra'` est une alternative plus courte pour la même localisation.

Ainsi, en ajustant la locale de cette manière, je m'assure que toutes les opérations temporelles, comme l'affichage de dates, suivront les conventions françaises. Cela contribue à rendre mon application plus conviviale et adaptée à l'audience francophone.

```
// Ajuste le locale en français
setlocale(LC_TIME, 'fr_FR.utf8','fra');
```

Après avoir extrais la date du message, qui est stockée sous forme de timestamp dans la variable `$contact['envoyé_le']`. En utilisant la fonction `strtotime`, je convertis ce timestamp en une représentation numérique du moment où le message a été envoyé.

```
// Convertit la date en string en timestamp
$timestamp = strtotime($contact['envoyé_le']);|
```

Ensuite, pour rendre cette information plus lisible, j'utilise la fonction `strftime` pour le convertir en une chaîne de caractères. Cela me permet de formater la date selon mes préférences, en l'occurrence, au format français avec le jour de la semaine, le jour du mois en numérique, le nom du mois et l'année sur 4 chiffres. Ainsi, mes utilisateurs peuvent facilement comprendre quand le message a été envoyé.

```
// Convertit le timestamp en date au format français et heure
// %A(Nom complet du jour de la semaine) / %d(Jour du mois en numérique, sur 2 chiffres (avec le zéro initial)) / %B(Nom complet du mois, suivant la locale) / %Y(L'année, sur 4 chiffres)
$date = strftime('%A %d %B %Y', $timestamp);
```

Aperçu :

DEMANDES DE CONTACT

| Envoyé le | Message |
|--------------------------------|-------------------------------------|
| lundi 13 novembre 2023 à 14:24 | Hello |
| lundi 13 novembre 2023 à 14:25 | UPDATE - Changement de Mot de Passe |

LEXIQUE

Responsive Design : Une approche de conception web qui garantit que l'interface utilisateur s'ajuste de manière optimale à différents dispositifs et tailles d'écrans, offrant une expérience utilisateur cohérente.

Backend : La partie d'une application qui gère les opérations en coulisses, généralement liée à la gestion de base de données et aux interactions avec le serveur.

Frontend : La partie d'une application avec laquelle l'utilisateur interagit directement, comprenant l'interface utilisateur et les fonctionnalités visibles.

CMS (Système de Gestion de Contenu) : Une plateforme logicielle permettant de créer, gérer et publier du contenu sur le web sans nécessiter de connaissances approfondies en programmation.

HTML (HyperText Markup Language): Langage de balisage utilisé pour structurer et décrire le contenu d'une page web.

Attribut : Propriété ajoutée à une balise HTML pour fournir des informations supplémentaires ou modifier son comportement.

Framework : Un ensemble de bibliothèques, d'outils et de conventions qui facilitent le développement d'applications en fournissant une structure de base et des fonctionnalités communes.

Bootstrap : Framework CSS pour la conception de sites web réactifs et adaptatifs.

jQuery : Bibliothèque JavaScript simplifiant la manipulation du DOM, l'animation et la gestion des événements.

CSS (Cascading Style Sheets) : Langage de feuille de style utilisé pour décrire l'apparence et la mise en forme d'un document HTML.

JavaScript (JS) : Langage de programmation utilisé pour ajouter de l'interactivité et des fonctionnalités avancées aux pages web.

Fonction JavaScript : Bloc de code réutilisable qui effectue une action spécifique.

Événement JavaScript : Action déclenchée par l'utilisateur ou le navigateur, comme un clic de souris ou le chargement d'une page.

JQuery : jQuery est une bibliothèque JavaScript pour simplifier la création de sites web interactifs.

PHP (Hypertext Preprocessor) : Langage de script côté serveur utilisé pour générer des pages web dynamiques et gérer les données côté serveur.

Variable PHP : Conteneur pour stocker des données en PHP.

Fonction PHP : Bloc de code réutilisable en PHP qui effectue une action spécifique.

MySQL (My Structured Query Language) : Système de gestion de base de données relationnelle utilisé pour stocker et gérer des données pour des applications web.

SQL (Structured Query Language) : Langage de programmation pour gérer et manipuler des bases de données relationnelles, comme MySQL, PostgreSQL, Oracle et SQL Server.

Base de données : Collection organisée de données.

Table MySQL : Structure dans une base de données MySQL pour stocker des données sous forme de lignes et de colonnes.

Requête SQL : Commande permettant d'interagir avec une base de données, comme insérer, récupérer, mettre à jour ou supprimer des données (CRUD : Create, Read, Update and Delete).

DOM (Document Object Model) : Représentation structurée d'une page web qui permet aux langages de script d'accéder et de modifier le contenu, la structure et le style de la page.

Superglobale : Variable prédéfinie en PHP qui contient des informations globales, telles que les données GET, POST et SESSION.

Session : Mécanisme permettant de stocker des données de manière persistante entre différentes pages d'une application web.

Cookie : Fichier texte stocké sur l'ordinateur de l'utilisateur, contenant des informations sur ce dernier et permettant de conserver des données entre les visites sur un site web.

PDO (PHP Data Objects) : Interface pour accéder aux bases de données en PHP, offrant une abstraction pour différentes bases de données et une sécurité accrue grâce aux requêtes préparées.

API (Interface de Programmation d'Application) : Ensemble de règles et de protocoles permettant à différents logiciels de communiquer entre eux. Les développeurs l'utilisent pour intégrer des fonctionnalités d'autres services dans leurs applications.

Git : Un système de contrôle de version qui permet de suivre les modifications apportées au code source d'un projet, facilitant le travail collaboratif et la gestion des versions.

CONCLUSION

En arrivant au terme de ce dossier projet, je ressens le besoin de réfléchir sur le chemin parcouru au cours de cette formation en développement web à la SJT.

Cette expérience ne se résume pas à une simple acquisition de compétences techniques. Elle représente une immersion profonde dans l'univers captivant du développement web, où chaque ligne de code écrite a contribué à l'édification de mes connaissances.

Au fil des semaines, j'ai découvert non seulement la complexité des langages de programmation et des technologies modernes, mais aussi la puissance de la résolution de problèmes, de la créativité et de la collaboration. Chaque projet a été une opportunité d'appliquer les connaissances acquises en classe, de résoudre des défis concrets et de perfectionner mes compétences.

Cette formation m'a également offert l'occasion de travailler sur des projets diversifiés, couvrant des aspects variés du développement web, de la conception de l'interface utilisateur à la gestion de bases de données. Ces projets pratiques ont constitué les fondements essentiels de mon apprentissage, me permettant de concrétiser des concepts théoriques en compétences opérationnelles.

Notre formatrice a joué un rôle crucial dans mon parcours. Son expertise, son dévouement et sa capacité à expliquer des concepts complexes ont été des éléments déclencheurs pour mon développement professionnel. Son accompagnement attentif a transformé des moments de perplexité en opportunités d'apprentissage.

En sortant de cette formation, je me sens armé de compétences solides, prêt à relever les défis du monde professionnel du développement web. Je suis désormais capable de concevoir des solutions innovantes, de créer des expériences utilisateur exceptionnelles et de contribuer de manière significative à des projets complexes.

En conclusion, cette formation n'a pas seulement été un chapitre de ma vie académique, mais une étape cruciale qui a façonné ma trajectoire professionnelle. Elle m'a fourni les outils nécessaires pour explorer le vaste paysage du développement web et a stimulé ma passion pour la création technologique.

Je tiens à exprimer ma gratitude envers tous ceux qui ont contribué à cette aventure – notre formatrice, les collègues et tous ceux qui ont partagé leurs connaissances. Je suis excité par les opportunités qui se présentent à moi et suis convaincu que les bases solides acquises au cours de cette formation seront la clé de mon succès futur.

Merci pour cette expérience enrichissante et pour l'opportunité de grandir en tant que développeur web.