**Dossier de Projet**

**Développeur Web et Web Mobile**

Une image contenant personne, Visage humain, jeune enfant, enfant

Description générée automatiquement

**Hatt’ Baby Haven.**

**LE PARADIS DES BÉbÉS.**

**Hattaoui El Mehdi | Promotion 2023**

**Sommaire.**

**Sommaire....................................................................................................................2**

**Compétences du référentiel couvertes par le projet.................................................3**

**Résumé du projet.......................................................................................................4**

**Cahier des charges......................................................................................................5**

**Contexte..................................................................................................................6**

**Utilisation................................................................................................................7**

**Spécifications techniques.........................................................................................8**

**Langages................................................................................................................9**

**Applications.........................................................................................................10**

**Autoloader.............................................................................................................11**

**Étape 1 : Configuration de .htaccess......................................................................12**

**Étape 2 : Index.php...............................................................................................13**

**Étape 3 : Autoloader.php.....................................................................................14**

**Réalisation : CRUD Utilisateur................................................................................15**

**CREATE : Inscription.........................................................................................16**

**READ : Mise à jour de la page "Mon compte".....................................................17**

**DELETE : Suppression de la page "Mon compte"...............................................18**

**Jeu d'essai...........................................................................................................19**

**Envoi des données avec Fetch.............................................................................20**

**Réception des données avec PHP.........................................................................21**

**Réception des données en JavaScript...................................................................22**

**Exemple.............................................................................................................23**

**Veille sur les vulnérabilités de sécurité.....................................................................24**

**Les injections SQL..............................................................................................26**

**Recherche sur un site anglophone...........................................................................27**

**Traduction........................................................................................................28**

**Conclusion......................................................................................................29**

**Compétences du référentiel couvertes par le projet.**

|  |  |  |
| --- | --- | --- |
| **ActivitÉ-type 1** **Développer la partie Front-end d’une application web et web mobile en intégrant les recommandations de sécurité.** | 1 | **Maquetter une application** |
| 2 | **Réaliser une interface utilisateur web statique et adaptable** |
| 3 | **Développer une interface utilisateur web dynamique** |
| 4 | Réaliser une interface utilisateur avec une solution de contenu ou e-commerce |
| **ActivitÉ-type 2** **Développer la partie Back-end d’une application web et web mobile en intégrant les recommandations de sécurité** | 5 | **Créer une base de données** |
| 6 | **Développer la partie back-end d’une application web et web mobile** |
| 7 | **Développer la partie back-end d’une application web et web mobile** |
| 8 | Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce |

**Résumé de projet.**

La box Hatt’s Baby Haven.

**Les parents sont de plus en plus soucieux de la santé et du bien-être de leurs enfants, et cela se reflète dans leurs choix en matière de produits cosmétiques. De nos jours, la demande croissante pour des produits plus naturels et respectueux de l’environnement a donné naissance à une tendance qui met l’accent sur les produits cosmétiques pour bébés et enfants.**

**La peau des bébés et des jeunes enfants est particulièrement délicate et sensible. Elle est plus mince et plus vulnérable aux irritations ainsi qu’à la sécheresse et aux réactions allergiques. Les produits cosmétiques conventionnels peuvent contenir des ingrédients potentiellement nocifs tels que certains conservateurs, ou des ingrédients desséchants, qui peuvent causer des problèmes cutanés. Les produits cosmétiques bio offrent une alternative plus douce, plus sûre et plus écologique.**

**C’est pourquoi Hatt’s Baby Haven, c’est le concept de box par abonnement qui permet de prendre soin de votre enfant et découvrir de belles marques et créateurs. Avec des produits de grande qualité et toujours bien pensés ! Leur contenu varie tous les mois pour s’adapter aux besoins et à l’évolution de votre tout petit.**

**Cahier des charges.**

**Erratum.**

**Malheureusement toutes les fonctionnalités présentes dans le résumé de projet n’ont pas pu être réalisées dans le temps.**

**Notamment : les liens relatifs à l’abonnement, des articles, des guides, des vidéos éducatives et des témoignages de parents.**

**Context.**

**Le projet Abonnement Bio Bébé vise à fournir aux parents une solution pratique et écologique pour répondre aux besoins essentiels de leurs bébés tout en favorisant des produits respectueux de l'environnement et de la santé de l'enfant.**

**Les parents cherchent des produits de haute qualité, sûrs et naturels pour leurs bébés, notamment des couches biodégradables, des aliments biologiques pour nourrissons et des produits de soin doux.**

**Flexibilité pour les parents car ils auront la possibilité de personnaliser leur abonnement en fonction des besoins spécifiques de leur bébé. Ils pourront choisir la fréquence de livraison, la taille des vêtements et des couches, ainsi que les produits spécifiques qu'ils souhaitent recevoir.**

**Ce projet vise à simplifier la vie des parents tout en garantissant la meilleure qualité de soins et de produits pour leurs bébés. Il répond aux besoins croissants de familles recherchant des solutions durables, biologiques et pratiques pour prendre soin de leur bébé.**

**Usage pour les utilisateurs non-inscrit.**

**Les utilisateurs non-inscrits peuvent accéder à une variété de contenus informatifs sur le site de puériculture sans avoir besoin de s'inscrire. Cela inclut des articles, des guides, des vidéos éducatives et des témoignages de parents. Ils peuvent ainsi se renseigner sur des sujets tels que la nutrition pour bébés, la sécurité des enfants, les étapes du développement, les meilleures pratiques pour l'allaitement, et bien d'autres. Ces informations sont utiles pour les futurs parents, les grands-parents, ou toute personne en quête de conseils sur la parentalité.**

**La recherche de produits et prix :**

**Les utilisateurs non-inscrits peuvent parcourir la gamme de produits disponibles sur le site de puériculture. Ils peuvent consulter les descriptions, les caractéristiques, les avis des clients et les prix des articles, ce qui les aide à prendre des décisions éclairées sur les produits dont ils pourraient avoir besoin. Cette fonctionnalité est particulièrement utile pour les personnes qui souhaitent comparer les offres avant de prendre une décision d'achat.**

**L’ajout d'Articles au panier :**

**Même sans inscription, les utilisateurs peuvent ajouter des articles à leur panier virtuel. Cela leur permet de préparer une liste de produits qu'ils envisagent d'acheter ultérieurement ou de partager avec d'autres membres de la famille pour obtenir leur avis. Cependant, pour finaliser l'achat, l'inscription peut être requise.**

**Une navigation facilitée :**

**Les utilisateurs non-inscrits bénéficient d'une navigation facilitée et d'une expérience conviviale sur le site. Ils peuvent explorer différentes catégories de produits, utiliser des filtres de recherche, et accéder à des outils pratiques, tels que des calculateurs de besoins pour bébés ou des listes de naissance préconçues.**

**Cela permet aux utilisateurs non-inscrits de profiter des informations et des avantages offerts par le site de puériculture tout en décidant s'ils souhaitent s'inscrire pour accéder à des fonctionnalités plus avancées, telles que la gestion de listes de naissance, des remises spéciales ou un suivi de commandes personnalisé.**

**Quand l’utilisateurs est inscrit.**

**Il peut :**

* **Ajoutez des produits à votre sélection en utilisant la fonction dédiée sur la page.**
* **Personnalisez ou retirez des articles et gérez les événements que vous avez ajoutés, notamment les rencontres entre parents.**
* **Mettez à jour les informations de votre compte en toute simplicité.**
* **Si vous le souhaitez, vous pouvez vous désinscrire.**
* **Terminez votre session en vous déconnectant.**

**Spécifications techniques.**

**Les langages et applications utilisés :**

* **HTML 5 « Hypertext Markup Language »**

 **HTML5 est la dernière version du langage de balisage HTML, utilisé pour créer et structurer des pages web.**

* **CSS (Cascading Style Sheets) :**

**CSS Beispiele [IBK Wiki]** **Les CSS sont un langage de feuilles de style utilisé pour définir la présentation et l'apparence des éléments HTML. Les CSS permettent de contrôler les polices, les couleurs, les marges, les dispositions, les animations, et d'autres aspects visuels d'un site web.**

* **JavaScript :**

 **JavaScript est un langage de programmation de script utilisé principalement dans le développement web pour rendre les pages web interactives et dynamiques. Il permet d'ajouter des fonctionnalités comme des diaporamas, des formulaires interactifs, des animations, des jeux, et des mises à jour en temps réel.**

* **PHP (Hypertext Preprocessor) :**

**PHP est un langage de programmation serveur largement utilisé pour le développement web dynamique. Il est principalement utilisé pour générer des pages web dynamiques en interagissant avec une base de données. PHP peut être incorporé dans le code HTML pour effectuer des tâches telles que la collecte de données de formulaires, la création de systèmes de gestion de contenu (CMS), la gestion de sessions utilisateur, et bien plus encore.**

* **SQL (Structured Query Language) :**

 **SQL est un langage de programmation spécialement conçu pour gérer et interagir avec les bases de données relationnelles. Il permet de créer, modifier, interroger et gérer des bases de données, y compris l'ajout, la suppression et la modification de données.**

* **Visual Studio Code :**

 **Visual Studio Code, souvent abrégé VS Code, est un éditeur de code source développé par Microsoft. C'est un outil populaire parmi les développeurs de logiciels car il offre un environnement de développement léger, personnalisable et extensible pour de nombreux langages de programmation. Il m’a permis d’écrire et de gérer mon code.**

* **PhpMyAdmin :**

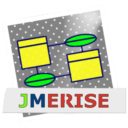
**PhpMyAdmin est une application web open source écrite en PHP et destinée à la gestion de bases de données MySQL. Elle offre une interface conviviale pour administrer des bases de données, ce qui inclut la création de tables, l'importation et l'exportation de données, la gestion des utilisateurs, l'exécution de requêtes SQL, et bien d'autres tâches liées à la gestion de bases de données MySQL. C'est un outil très utile pour les administrateurs de bases de données et les développeurs web.**



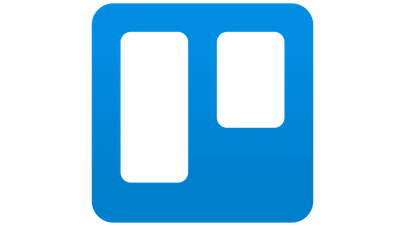
[Cette photo](https://www.pngall.com/phpmyadmin-png/download/67489) par Auteur inconnu est soumise à la licence [CC BY-NC](https://creativecommons.org/licenses/by-nc/3.0/)

* **Figma**

 **Figma est une application de conception d'interface utilisateur (UI) et d'expérience utilisateur (UX) basée sur le cloud, largement utilisée par les concepteurs et les équipes de développement de logiciels. Elle permet de créer, de collaborer et de prototyper des designs interactifs pour des applications, des sites web, des applications mobiles et d'autres produits numériques. Il m’a servi pour le maquettage de mon site**

* **JMerise**

**C’est une application dédiée à la modélisation des modèles conceptuels de base de données (MCS), qui génère le modèle logique de base de données (MLD) ainsi que le script SQL.**

* **Trello**

**J'ai utilisé Trello, une application web de gestion de projet, pour par la suite créer mon cahier des charges du site en utilisant la méthode des "user stories".**

* **Gloomaps**

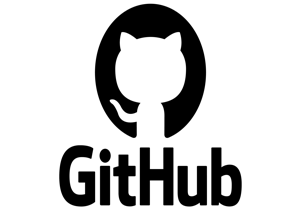
Une image contenant Police, Graphique, logo, capture d’écran

Description générée automatiquement

**Gloomaps sert à la création de l’arborescence du site, représentation des différentes pages et fonctionnalités nécessaires.**

* **GIT & GITHUB**



**Git est un système de gestion de version distribuée, largement utilisé dans le développement de logiciels, qui permet de suivre et de gérer les modifications apportées au code source d'un projet.**

**GitHub est une plateforme web d'hébergement de projets Git. Elle permet aux développeurs de stocker, de partager, de collaborer et de gérer des projets de développement de logiciels**

**Cela m’a servi pour faire des sauvegardes régulières de mon projet.**

**Maquettage du site web et rendu.**

**Arborescence et plan du site Hatt’s Baby Haven.Une image contenant capture d’écran, texte

Description générée automatiquement**

**Avant la réalisation du projet, j’ai réalisé un maquettage de mon site Internet afin d’avoir un visuel avec les grandes idées. J’ai alors commencé par faire le un zoning sur des feuilles de papier comme brouillon puis par la suite j’ai enfin créé une maquette graphique à l’aide de Figma pour avoir un aperçu du futur projet.**

**Une image contenant capture d’écran, Rectangle, texte, conception

Description générée automatiquement**

Une image contenant capture d’écran, Rectangle, conception, distributeur de billets

Description générée automatiquement

Une image contenant capture d’écran, Rectangle, conception, électroménager

Description générée automatiquement

Une image contenant Rectangle, capture d’écran, carré, conception

Description générée automatiquement

**Réalisation : CRUD Utilisateur.**

**CREATE – Inscription**

**Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquementImage du formulaire d’inscription sur le site Hatt’s Baby Haven.**

**Ce formulaire d'inscription présente plusieurs fonctionnalités remarquables. Chacun de ses champs est soumis à une vérification en temps réel à l'aide de regex spécifiques. Chaque fois qu'un champ est renseigné, le système effectue une vérification instantanée pour déterminer si la valeur est correcte ou non.**

**En cas d'erreur, un message d'erreur informatif s'affiche en rouge, guidant ainsi l'utilisateur.**

**Ces vérifications sont réalisées à l'aide d'expressions régulières adaptées aux données attendues, garantissant ainsi une saisie précise et conforme aux critères prédéfinis.**

**Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquementImportation des fichiers et déclaration des regex du fichier inscription.**

**1. session\_start(): Cette fonction est utilisée pour démarrer ou reprendre une session utilisateur. Les sessions permettent de stocker des données persistantes d'une page à l'autre, ce qui est utile pour conserver des informations d'état d'utilisateur, telles que les informations de connexion.**

**2. require\_once '../errors.php': Cette ligne inclut un fichier nommé 'errors.php' depuis un répertoire parent (indiqué par '..'). Ce fichier peut contenir des définitions d'erreurs, des messages d'erreur ou d'autres éléments liés à la gestion des erreurs dans l'application. Le require\_once sert à importer un ou plusieurs fichiers.**

**4. $formErrors = []: Cette ligne initialise un tableau vide appelé $formErrors. Ce tableau est généralement utilisé pour stocker des erreurs ou des messages liés aux formulaires que les utilisateurs soumettent. Il est probable que ce tableau sera rempli au fur et à mesure que le script traitera les données de formulaire et les validera.**

**5. Pour les $regex lors de l’envoi du formulaire, c’est cette fois une classe PHP qui est appelée à vérifier les données envoyées. Celles-ci vérifient que les champs ne sont pas vides et que les données envoyées correspondent à des regex spécifiques.**

**L'expression 'name' permet de valider des noms ou prénoms avec des lettres, y compris les caractères accentués.**

**L'expression 'birthdate' vérifie si la date de naissance est bien au format "AAAA-MM-JJ".**

**L'expression 'password' sert à valider les mots de passe forts en imposant des critères : au moins une lettre majuscule ou minuscule, un chiffre, et un caractère spécial, avec une longueur minimale de 8 caractères.**

**Les regex servent à valider des données telles que les noms, les dates de naissance et les mots de passe, en s'assurant qu'elles correspondent à un format ou à des critères spécifiques.**

**Je vérifie si la donnée n’est pas vide est quelle correspond à la regex**

**Si c’est le cas je stocke cet donnés dans la propriété firstname de l’objet user**

**Pour ce faire, les informations sont envoyées en mode POST à la classe User Controller et sont ensuite traitées au moyen de la méthode preg\_match.**

**Une image contenant texte, Police, capture d’écran

Description générée automatiquementImage de processus de validation des données utilisateurs du formulaire d’inscription**

**Ce code PHP fait partie d'un processus de validation de données utilisateur, du formulaire d'inscription ou de mise à jour de profil. Voici une explication de ce code :**

**Pour la Validation du Prénom (firstname) :**

**Il vérifie d'abord si le champ "firstname" dans les données POST n'est pas vide (! empty($\_POST['firstname'])).**

**Ensuite, il utilise la fonction preg\_match pour comparer la valeur du champ "firstname" avec une expression régulière stockée dans $regex['name'].**

**Si la valeur du champ "firstname" correspond à l'expression régulière (c'est-à-dire qu'elle satisfait aux critères définis), alors il nettoie la valeur en supprimant toute balise HTML potentiellement dangereuse avec strip\_tags, et assigne cette valeur à la propriété $user->firstname.**

**Sinon, s'il y a une incompatibilité avec l'expression régulière, il stocke un message d'erreur dans le tableau $formErrors associé au champ "firstname".**

**Pour la Validation du Nom (lastname) :**

**Il suit le même processus que le champ "firstname"**

**Une image contenant texte, Police, capture d’écran, conception

Description générée automatiquementPour la Vérifier l’email :**

**Image de la méthode checkAvaibility dans userModel.php**

**Pour cette méthode j’utilise la fonction checkAvaibility dans mon userModel.php, elle vérifie si l’adresse email spécifiée existe déjà dans une table de base de données. Elle compte combien de fois cette adresse email est présente en utilisant une requête SQL préparée pour éviter les attaques par injection.**

**La méthode renvoie le nombre d'occurrences de l'adresse e-mail dans la base de données, permettant ainsi de vérifier sa disponibilité (zéro occurrence) ou son utilisation.**

**Une image contenant texte, capture d’écran, Police

Description générée automatiquementImage des vérifications de l’email dans le registrerController.php**

**Ce code vérifie si l'adresse email entrée par un utilisateur dans un formulaire est valide et non vide. Voici comment cela fonctionne de manière simple :**

**Il vérifie d'abord si le champ "email" dans les données POST n'est pas vide (!empty($\_POST['email'])). Cela signifie que l'utilisateur a bien saisi quelque chose dans le champ.**

**J’appelle ma méthode checkAvaibility de mon modèle user, si la méthode est égale à 1, autrement dit si l’email existe déjà dans la base de données, j’affiche l’erreur correspondante**

**Ensuite, il utilise la fonction filter\_var avec l'option FILTER\_VALIDATE\_EMAIL pour vérifier si ce que l'utilisateur a saisi ressemble à une adresse email valide.**

**Si l'adresse email est valide et non vide, alors l'adresse e-mail est considérée comme correcte. Dans ce cas, l'adresse email est nettoyée (les balises HTML potentiellement dangereuses sont supprimées) et ensuite assignée à la propriété email de l'objet $user.**

**Si l'adresse email n'est pas valide (par exemple, si elle ne contient pas "@" et d'autres éléments nécessaires pour une adresse email valide) ou si elle est vide, un message d'erreur spécifique est enregistré dans un tableau appelé $formErrors. Ce message d'erreur ("USERS\_EMAIL\_ERROR\_INVALID") signale à l'utilisateur que l'adresse email saisie n'est pas correcte. Si le champ est laissé vide, un message d'erreur différent ("USERS\_EMAIL\_ERROR\_EMPTY") est stocké pour indiquer que l'adresse email est nécessaire.**

**En résumé, ce code vérifie si l'adresse email saisie est valide et non vide, la nettoie si c'est le cas, et enregistre un message d'erreur si les conditions ne sont pas remplies.**

**Les erreurs :**

**("USERS\_FIRSTNAME\_ERROR\_INVALID", "USERS\_FIRSTNAME\_ERROR\_EMPTY", "USERS\_LASTNAME\_ERROR\_INVALID", "USERS\_LASTNAME\_ERROR\_EMPTY") sont des constantes qui décrivent les problèmes spécifiques rencontrés lors de la validation des données.**

**Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement**

**Image de form errors de la classe firstname dans registrerController.php**

**Une image contenant texte, Police, capture d’écran, conception

Description générée automatiquement**

**Image des constantes errors dans fichier errors.php exemple pour firstname**

**Les messages d'erreur quant à eux stockés dans $formErrors je les ai définies dans le fichier errors.php pour ensuite les appeler dans le fichier Controller.**

**En résumé, ce code PHP assure la validation des champs "firstname" et "lastname" d'un formulaire. Si les données ne respectent pas les critères définis dans les expressions régulières, des messages d'erreur appropriés sont enregistrés pour informer l'utilisateur des erreurs dans son entré**

**Une image contenant texte, Police, capture d’écran, conception

Description générée automatiquementDans mon usermodel.php j’utilise la méthode add pour ajouter un utilisateur à la base de données.**

**Image de la méthode add dans userModel.php**

**La méthode add est définie, ce qui indique qu'elle sera utilisée pour ajouter un utilisateur à la base de données.**

**Une requête SQL est définie dans la variable $query. Cette requête d'insertion (INSERT INTO) est destinée à ajouter une nouvelle ligne (un nouvel utilisateur) dans la table de base de données kh9jg\_users. Les champs de cette table sont lastname, firstname, birthdate, email, et password.**

**La requête SQL est préparée à l'aide de la méthode prepare de l'objet PDO ($this->db->prepare()). La préparation de la requête est une pratique recommandée pour éviter les attaques par injection SQL et garantir la sécurité des données.**

**Les valeurs à insérer dans la requête sont associées aux marqueurs nominatifs (par exemple, : lastname, :firstname) à l'aide de la méthode bindValue. Cela signifie que les valeurs des propriétés de l'objet courant seront insérées dans la requête.**

**Les marqueurs nominatifs sont associés aux propriétés de l'objet utilisateur ($this->lastname, $this->firstname, etc.) en utilisant bindValue. Les marqueurs et les propriétés doivent correspondre.**

**Enfin, la requête est exécutée à l'aide de la méthode execute(), et la méthode add retourne le résultat de cette exécution. Si l'insertion est réussie, execute() renverra true. Si l'insertion échoue, elle renverra false.**

**Ce code résume le processus d'ajout d'un nouvel utilisateur dans une base de données en utilisant une requête préparée pour garantir la sécurité des données et l'intégrité de la base de données.**

**Réalisation : CRUD Utilisateur.**

**READ – ‘Mon compte’**

**Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquementImage page mon compte**

**La page « Mon compte » quant à elle regroupe plusieurs fonctionnalités.**

**Ici sont affichés les informations de l’utilisateur, les champs sont cliquables et permettent de changer ses informations facilement.**

**Comme montré précédemment pour l’inscription , les champs ont aussi une vérification.**

**Image mon compte modification des informations**

Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement

**La vérification PHP permet aussi de vérifier qu’un champ contient bien une nouvelle valeur et non la même que précédemment. Ces valeurs à modifier sont ajoutées à une propriété tableau’ ‘.**

Une image contenant texte, Police, capture d’écran

Description générée automatiquement**Image des vérifications de l’email dans le profilController.php**

**J’effectue la validation d'une adresse e-mail soumise via un formulaire, en s'assurant que l'adresse est valide et disponible. Voici comment cela fonctionne en termes simples :**

**je vérifie d'abord si le champ "email" dans les données POST n'est pas vide (!empty($\_POST['email'])). Cela signifie que l'utilisateur a saisi quelque chose dans le champ.**

**Ensuite, j’ utilise la fonction filter\_var avec l'option FILTER\_VALIDATE\_EMAIL pour vérifier si ce que l'utilisateur a saisi ressemble à une adresse e-mail valide. Si c'est le cas, l'adresse e-mail est nettoyée (les balises HTML potentiellement dangereuses sont supprimées) et assignée à la propriété email de l'objet $userUpdate.**

**Je contrôle également si l'adresse e-mail est déjà utilisée par un autre utilisateur dans la base de données. Si c'est le cas et que l'adresse e-mail soumise n'est pas celle de l'utilisateur actuellement connecté (stockée dans la session), une erreur est enregistrée.**

**Si l'adresse e-mail n'est pas valide (par exemple, si elle ne contient pas "@" et d'autres éléments nécessaires pour une adresse e-mail valide) ou si elle est vide, un message d'erreur spécifique est stocké dans un tableau appelé $formErrors. Ce message d'erreur ("USERS\_EMAIL\_ERROR\_INVALID") signale à l'utilisateur que l'adresse e-mail saisie n'est pas correcte. Si le champ est laissé vide, un message d'erreur différent ("USERS\_EMAIL\_ERROR\_EMPTY") est stocké pour indiquer que l'adresse e-mail est nécessaire.**

**En résumé, cette méthode vérifie la validité et la disponibilité d'une adresse e-mail soumise par un utilisateur dans un formulaire. Il nettoie l'adresse e-mail, la compare à celles déjà existantes dans la base de données, et stocke des messages d'erreur appropriés en cas de problème.**

**Réalisation : CRUD Utilisateur.**

Une image contenant texte, capture d’écran, Police

Description générée automatiquement**UPDATE - ‘Mon compte’**

**Image de la méthode de mise à jour profils sur userModel.php**

**J’utilise une méthode pour mettre à jour les informations d'un utilisateur dans une base de données. Voici comment cela fonctionne :**

**public function updateInfos(): Cette ligne déclare une méthode publique nommée "updateInfos" qui sera utilisée pour mettre à jour les informations d'un utilisateur.**

**La requête SQL est définie dans la variable $query. Cette requête est une instruction UPDATE qui indique quels champs de la table kh9jg\_users doivent être modifiés. Les champs à mettre à jour sont lastname, firstname, birthdate, et email. La clause WHERE spécifie que la mise à jour doit s'appliquer à la ligne correspondant à un identifiant (id) spécifique.**

**La requête SQL est préparée à l'aide de la méthode prepare de l'objet PDO. La préparation de la requête est une pratique recommandée pour éviter les attaques par injection SQL et garantir la sécurité des données.**

**Les valeurs à insérer dans la requête sont associées aux marqueurs nominatifs (:lastname, :firstname, :birthdate, :email, :id) à l'aide de la méthode bindValue. Cela signifie que les valeurs des propriétés de l'objet courant (probablement un modèle d'utilisateur) seront insérées dans la requête pour effectuer la mise à jour.**

**Les types de données des valeurs sont spécifiés (PDO::PARAM\_STR pour les chaînes de caractères et PDO::PARAM\_INT pour les entiers) pour garantir que les données sont correctement formatées.**

**Enfin, la requête est exécutée à l'aide de la méthode execute(), et la méthode updateInfos retourne le résultat de cette exécution. Si la mise à jour est réussie, execute() renverra true. Si la mise à jour échoue, elle renverra false.**

**En résumé, cette méthode permet de mettre à jour les informations d'un utilisateur dans la base de données en modifiant les champs spécifiés (nom, prénom, date de naissance et e-mail) pour l'utilisateur correspondant à un identifiant donné.**

**Réalisation : CRUD Utilisateur.**

**DELETE – ‘Supprimer mon compte’**