**Rapport de stage**

Tout d’abord, j’ai été chargé de créer des affiches pouvant servir dans l’enceinte du camping. **(Voir annexe 1)**

Ensuite, j'ai fait une analyse du référencement du Camping Les Portes de la baie de Somme. J'ai aussi travaillé sur la réalisation d'un tableau Excel répertoriant tous les campings situés dans un rayon de 20 km. **(Voir annexe 2)**

Plus tard, j’ai travaillé sur la réalisation d'un benchmark, en suivant un exemple donné par mon maître de stage. Le benchmarking est une technique qui consiste à comparer les performances, les pratiques et les stratégies d'une entreprise avec celles de ses concurrents ou d'autres entreprises du même secteur. L'objectif est d'identifier les meilleures pratiques et de déterminer les opportunités d'amélioration. Au fur et à mesure que j'avançais dans le benchmark, j'ai noté les forces et les faiblesses de chaque concurrent, ainsi que les pratiques exemplaires que je pourrais recommander à mon entreprise. J'ai également identifié des opportunités d'amélioration pour notre entreprise en comparant nos performances à celles des concurrents. **(Voir annexe 3)**

Pendant mon stage j’ai aussi approfondi mes connaissances sur WordPress. J'ai appris à naviguer dans l'interface et à utiliser les différentes fonctionnalités offertes par WordPress, telles que la gestion des pages, des articles, des plugins et des thèmes. J'ai également pu expérimenter avec la personnalisation du design en utilisant les options de thème disponibles. J'ai pu faire la création d'un site vitrine pour le camping. **(Voir annexe 4)**

En travaillant dans l'accueil, j'ai constaté que le gérant du camping n'était pas toujours disponible pour accueillir les clients. Cela m'a amené à assumer une partie de ces responsabilités en accueillant les clients et en leur fournissant les informations nécessaires.

Durant toute une semaine, j'ai consacré mon temps à la modification d'un template pour le site web du camping. Mon objectif était de personnaliser le template donné en modifiant les logos, les textes, les liens, les couleurs et les fichiers afin de créer un site qui représente au maximum l'état d'esprit et l'identité du camping. **(Voir annexe 5)**

Après avoir achevé cette étape, je me suis retrouvé à un point où je me demandais quelles améliorations supplémentaires je pourrais apporter au site pour améliorer l'expérience des visiteurs. J'ai donc eu une idée et j'en ai discuté avec mon maître de stage : pourquoi ne pas mettre en place un système de réservation en ligne qui permettrait aux visiteurs de choisir les éléments qu'ils souhaitent lors de leur séjour au camping en ligne ? J'ai expliqué mon idée en détail, suggérant que nous pouvions créer une fonctionnalité qui permettrait aux visiteurs de faire des réservations directement sur le site. Ils pourraient sélectionner les éléments spécifiques qu'ils désirent, comme le type d'hébergement, les dates de séjour, les services supplémentaires, etc. Ensuite, le gérant du camping aurait la possibilité d'accepter ou de refuser ces demandes de réservation, et il pourrait également contacter les visiteurs si des informations supplémentaires sont nécessaires.

Pour mettre en œuvre cette idée, j'ai créé une page d'inscription et de connexion où les visiteurs pourraient créer un compte et accéder à une page pour effectuer des réservations. Il m'a ensuite donné des conseils et des directives sur la manière de procéder, et nous avons fixé des objectifs pour les prochains jours.

J'ai également travaillé sur la création de la base de données correspondante. J'ai créé une table nommée "user" avec des colonnes telles que l'identifiant (id), le nom (name), le mot de passe (password), l'adresse e-mail (mail), le numéro de téléphone (phone) et le nombre de tentatives de connexion (login\_attempts). La colonne "login\_attempts" servira à compter les tentatives de connexion infructueuses afin de surveiller le trafic et d'éviter les attaques par force brute.

Pour l'inscription, le site demande aux visiteurs leur nom et prénom, ainsi que leur adresse e-mail qu'ils devaient entrer deux fois pour la confirmer. L'adresse e-mail jouerait un rôle important à la fois pour la connexion des utilisateurs et pour les contacter ultérieurement. J'ai également mis en place un système où les utilisateurs devaient entrer leur mot de passe deux fois pour le confirmer, afin d'éviter les erreurs de frappe. Enfin, j'ai inclus un champ pour le numéro de téléphone des utilisateurs, au cas où il serait nécessaire de les contacter par ce moyen.

J’ai appris l'importance de l'organisation du code. J'ai pris conscience qu'un code bien structuré et organisé facilite la compréhension du code. J'ai donc commencé à appliquer des pratiques de rangement dans mon code.

J’ai ensuite entamé le développement de la page de connexion pour le site du camping. Cette page permettrait aux utilisateurs enregistrés de se connecter en utilisant leur adresse e-mail et leur mot de passe. De plus, j'ai envisagé d'implémenter une fonctionnalité de blocage temporaire pour les utilisateurs qui entreraient de manière répétée un mot de passe incorrect. Cette mesure de sécurité vise à protéger les comptes des utilisateurs contre les attaques par force brute.

J'ai terminé la mise en place de la page de connexion, ce qui permettait aux utilisateurs enregistrés d'accéder à une zone réservée pour effectuer des réservations. Une fois connectés, ils étaient redirigés vers une page de réservation où ils pouvaient choisir leurs dates d'arrivée et de départ, ainsi que le type d'emplacement souhaité (mobil-home, tente/caravane ou camping-car) et le nombre total de personnes. La partie la plus délicate de cette journée a été la mise en place d'une fonctionnalité en JavaScript pour modifier le nombre maximal de personnes en fonction du type d'emplacement choisi. En effet, le nombre maximal de personnes autorisées dans un mobil-home était de 6, tandis que pour les autres types d'emplacements, il était de 10. J'ai dû me renseigner sur internet et explorer différentes méthodes pour trouver celle qui répondrait le mieux à mes attentes. Malgré quelques difficultés dues à ma compréhension encore limitée de JavaScript, j'ai finalement réussi à implémenter cette fonctionnalité avec succès. J'ai mis en place un code qui permettait de mettre à jour dynamiquement le nombre maximal de personnes autorisées en fonction du type d'emplacement sélectionné par l'utilisateur. Cette étape a été une occasion d'apprentissage et de développement personnel, car elle m'a permis d'améliorer ma compréhension et mes compétences en JavaScript.

Plus tard, j’ai ajouté des fonctionnalités essentielles pour améliorer l'expérience utilisateur. Tout d'abord, j'ai utilisé du JavaScript pour calculer le nombre de nuits en fonction des dates d'arrivée et de départ sélectionnées par les utilisateurs. En soustrayant la date de départ par la date d'arrivée, j'ai pu obtenir le nombre total de nuits de séjour. J'ai ensuite affiché cette information sur la page de réservation. **(Voir annexe 6)**

Ensuite, j'ai travaillé sur la création d'une fonction permettant de calculer le prix total de la réservation en fonction des différentes options sélectionnées par les utilisateurs. Cela incluait les dates de séjour, le type d'emplacement (mobil-home, tente/caravane, camping-car), le nombre de personnes adultes, le nombre d'enfants de moins de 12 ans/de plus de 12 ans, l’électricité, le nombre d’animaux et l’emplacement véhicule. **(Voir annexe 7)** Pour gérer le nombre d'enfants de moins de 12 ans, j'ai dû mettre en place des conditions spécifiques en fonction du nombre total de personnes sélectionnées. Par exemple, si une seule personne était choisie, aucun enfant ne pouvait être ajouté car il devait y avoir au moins un adulte accompagnant. De plus, j'ai veillé à ce que le nombre d'enfants ne dépasse jamais le nombre total de personnes et soit toujours accompagné d’un adulte.

Ensuite, j'ai créé une nouvelle table dans ma base de données appelée "reservation" pour stocker toutes les informations relatives à la réservation. Cette table comporte 11 colonnes, dont l'identifiant (id), l'identifiant de l'utilisateur (user\_id), le type d'emplacement (typeEmplacement), le nombre total de personnes (nbrPersonnes), le nombre d'enfants de moins de 12 ans (nbrEnfants), la demande d'électricité (electricite), la présence d'un véhicule (vehicule), le nombre d'animaux de compagnie (nbrAnimaux), la date d'entrée (dateEntree), la date de sortie (dateSortie) et le prix total de la réservation (prix). **(Voir annexe 8)** Pour lier ces informations à l'utilisateur, j'ai créé une relation entre les tables "user" et "reservation" en utilisant l'identifiant de l'utilisateur (user\_id) dans la table de réservation. Cela m'a permis d'associer chaque réservation à l'utilisateur correspondant.

Par la suite, j'ai commencé à mettre en place une interface de connexion pour les administrateurs. Dans la table "user", j'ai ajouté une colonne "isAdmin" qui a une valeur par défaut de "0" pour les utilisateurs non administrateurs, et une valeur de "1" pour les utilisateurs ayant des privilèges d'administration. Lorsqu'un utilisateur se connecte, le système vérifie si cet utilisateur est un administrateur. Si c'est le cas, il est redirigé vers une page de gestion des réservations, où il pourra visualiser et gérer toutes les réservations effectuées.

Pour commencer, j'ai utilisé un template de page avec un menu de navigation, afin de faciliter la navigation entre les différentes sections de la page. J'ai personnalisé ce template en remplaçant les structures existantes par celles qui correspondaient à mes besoins spécifiques. Pour le moment, je me suis concentré sur la création d'un bouton permettant de rediriger vers une autre page, celle dédiée à la gestion des demandes de réservations.

Dans cette page, j'ai travaillé sur l'affichage des demandes stockées dans la base de données. J'ai d'abord recherché et trouvé un autre template adapté à mes besoins, que j'ai modifié pour répondre à mes attentes. Ensuite, j'ai utilisé une requête SQL dans mon code pour sélectionner toutes les réservations dans la base de données. Pour chaque demande de réservation, j'ai également effectué d'autres requêtes SQL pour récupérer les informations relatives à l'utilisateur qui a fait la demande, telles que son nom, son adresse e-mail et son numéro de téléphone. J'ai affiché ces informations sur la page de gestion des réservations. Enfin, j'ai ajouté deux boutons supplémentaires : un pour accepter la demande de réservation et un autre pour la refuser. **(Voir annexe 9)**

Au vingt-septième jour de mon stage, j'ai apporté des modifications à ma page de réservation. Il me reste encore deux éléments à finaliser. Tout d'abord, je dois implémenter la déduction d'un jour au septième jour de réservation, car il est offert aux clients. Cela nécessitera une modification dans le calcul des dates. Ensuite, je dois ajuster le prix des mobil-homes en fonction des mois, car les tarifs varient selon la période de l'année. Cette tâche peut être plus complexe, car elle implique de prendre en compte les différentes saisons et d'adapter les prix en conséquence.

J'ai ensuite créé deux nouvelles pages : la première permet de visualiser les demandes de réservation acceptées, la seconde affiche les demandes refusées. Les demandes refusées peuvent être acceptées en cas de fausse manipulation, et vice versa pour les demandes acceptées. De plus, une limite de 3 réservations par jour par utilisateur a été ajouté pour éviter les abus.

J'ai réalisé plusieurs améliorations significatives dans le système de gestion des réservations pour les administrateurs. J'ai ajouté deux nouveaux boutons dans le menu des administrateurs, l'un permettant de supprimer toutes les réservations refusées et l'autre de supprimer toutes les réservations datant de plus de 2 mois. **(Voir annexe 10)**

J'ai également créé une nouvelle page dédiée à la création de réservations par les administrateurs. Contrairement aux visiteurs non administrateurs, les administrateurs ont désormais la possibilité de personnaliser les données de réservation à leur convenance. Ils peuvent entrer manuellement le nombre de personnes, le nombre de véhicules, le numéro d'emplacement, le nom et prénom de la personne, l'adresse e-mail et le numéro de téléphone. Ces informations personnelles sont stockées dans une nouvelle table de la base de données, associées à un compte utilisateur ayant un ID égal à 0. Aucun utilisateur ne peut se connecter à ce compte, et il sert uniquement à stocker les données des réservations créées par les administrateurs. **(Voir annexe 11)** De plus, j'ai ajouté un nouveau choix lors de la création de réservation pour les administrateurs. Ce nouveau choix permet d'accepter directement la réservation ou de la mettre en attente pour une validation ultérieure. Cette fonctionnalité permet aux administrateurs de créer des brouillons ou de prendre des notes pour les réservations en attente de décision. En cas d'erreur, si la date d'entrée est supérieure à la date de sortie, un message d'erreur approprié s'affiche pour guider les administrateurs et éviter les incohérences.

J’ai aussi créé une page pour modifier toutes les réservations. **(Voir annexe 12)**

J'ai commencé ensuite à créer la page du calendrier. Je souhaitais afficher les réservations acceptées de manière visuelle. J'ai conçu un affichage sous forme de calendrier, avec la possibilité de naviguer d'un mois à l'autre à l'aide de boutons "mois suivant" et "mois précédent". J'ai veillé à ce que la page n'affiche qu'un seul mois à la fois pour une meilleure lisibilité. La navigation est infinie. Pour l'affichage des réservations sur le calendrier, j'ai mis en place une logique qui regroupe les réservations d'un même emplacement sur une seule ligne horizontale. Par exemple, si l'emplacement M4 a trois réservations, celles-ci seront affichées sur la même ligne. J'ai également pris en compte les cas où les réservations se superposent en termes de dates. **(Voir annexe 13)** Si deux réservations se chevauchent, les cases correspondantes sur le calendrier sont affichées en rouge, sinon elles sont affichées en vert. (Voir annexe 13)

**Estimation et couts :**

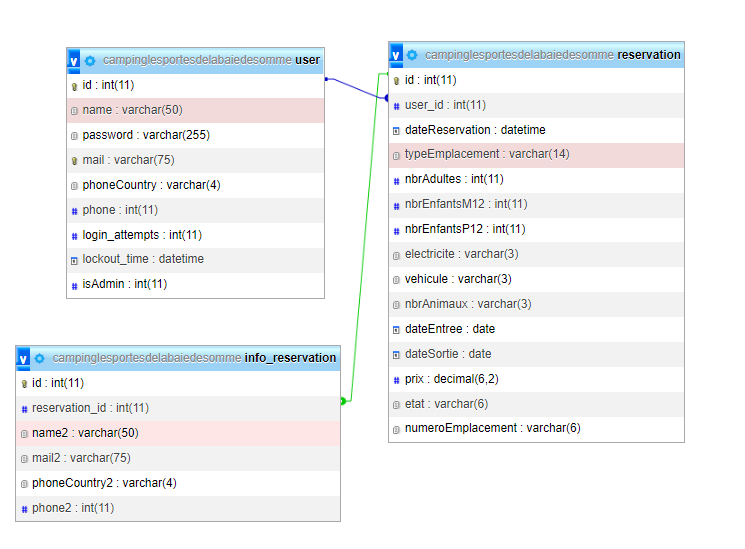
Salaire : 1 680€ brut / 1 344€ net

Couts : logiciel Photoshop 23,99 €/mois, PC d’environ 300€ minimum, logiciel de codage gratuit

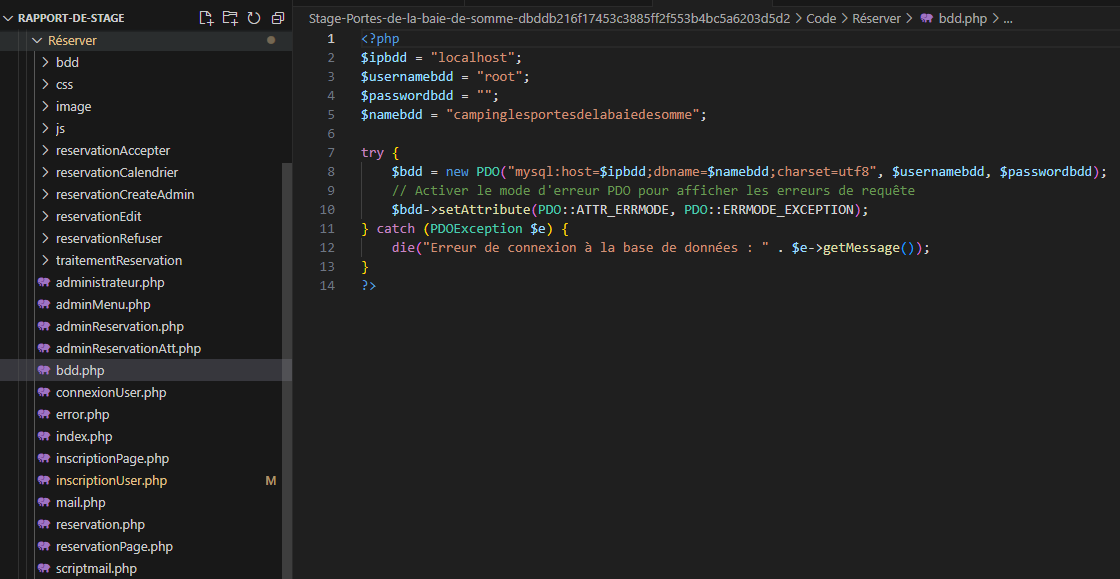
---> 347€

**Code :**

Voici la BDD :



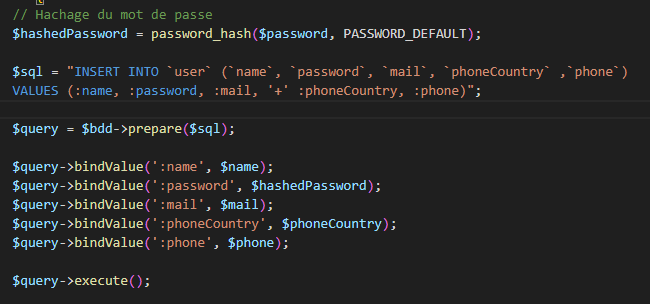
Tout d’abord, j’ai créé un fichier PHP nommé « bdd.php » pour la connexion à la BDD



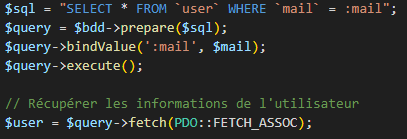
Ensuite, pour s’inscrire j’ai créé un formulaire qui possède la méthode POST.



Du code vérifie si le formulaire est soumis

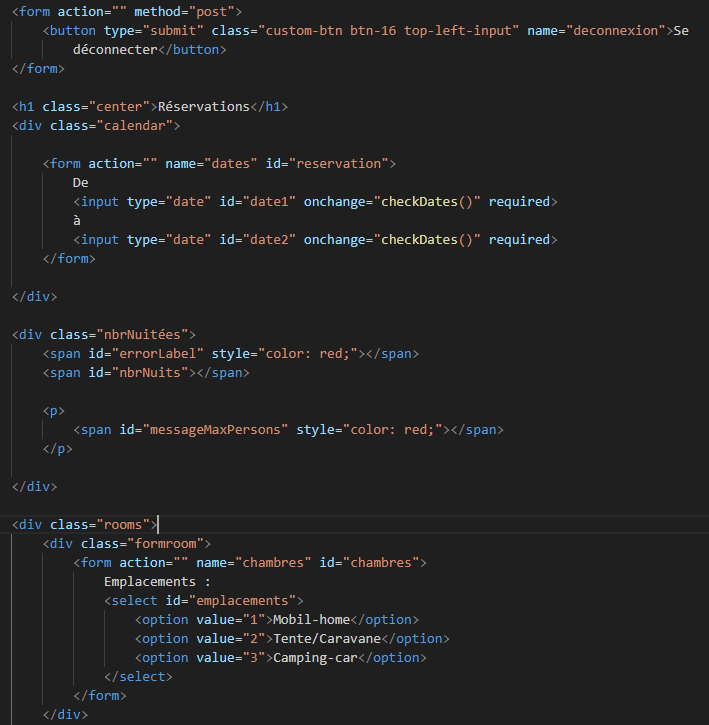
Si tout est ok, alors l’insertion se fait avec un hachage du mot de passe

Pour la connexion, un formulaire POST est aussi utilisé. Quand il est soumis, une requête SQL vérifie l’authentification. Si elle réussit, alors l’utilisateur est redirigé vers la page de réservation.

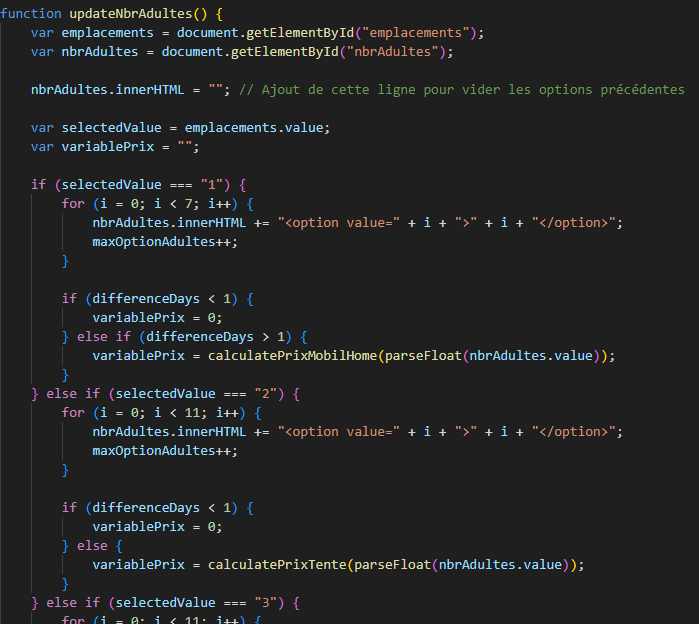




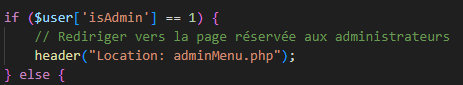
La page de réservation contient principalement du JavaScript. Un formulaire en méthode POST pour se déconnecter est disponible. Il y a majoritairement des formulaires liés à du code JS pour gérer des variables selon l’option sélectionnée.



Une fonction est créée afin de gérer le nombre d’adultes maximum par rapport à l’emplacement choisi.



Pour les administrateurs, quand ils se connecter, une requête SQL vérifie si isAdmin=1.



**Annexe :**

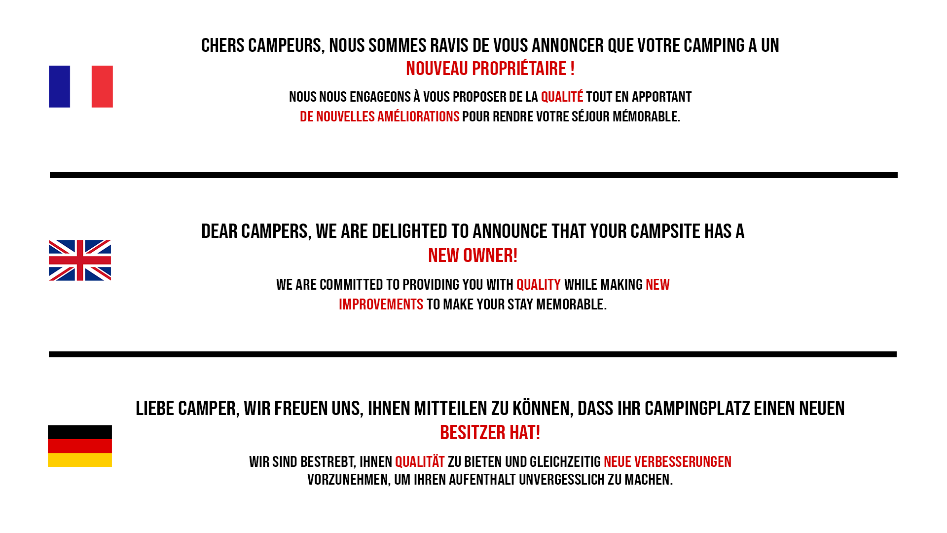
**Annexe 1 :**













**Annexe 2 :**



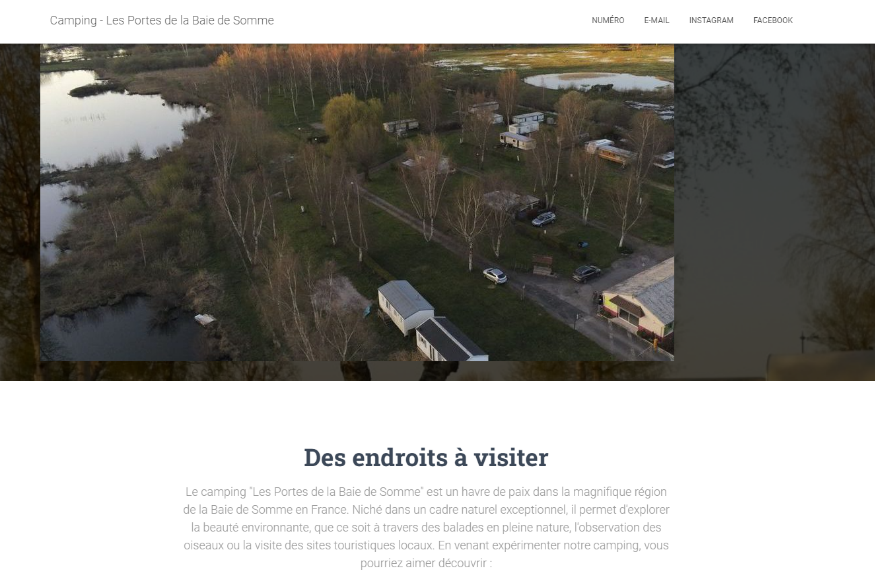
**Annexe 3 :**





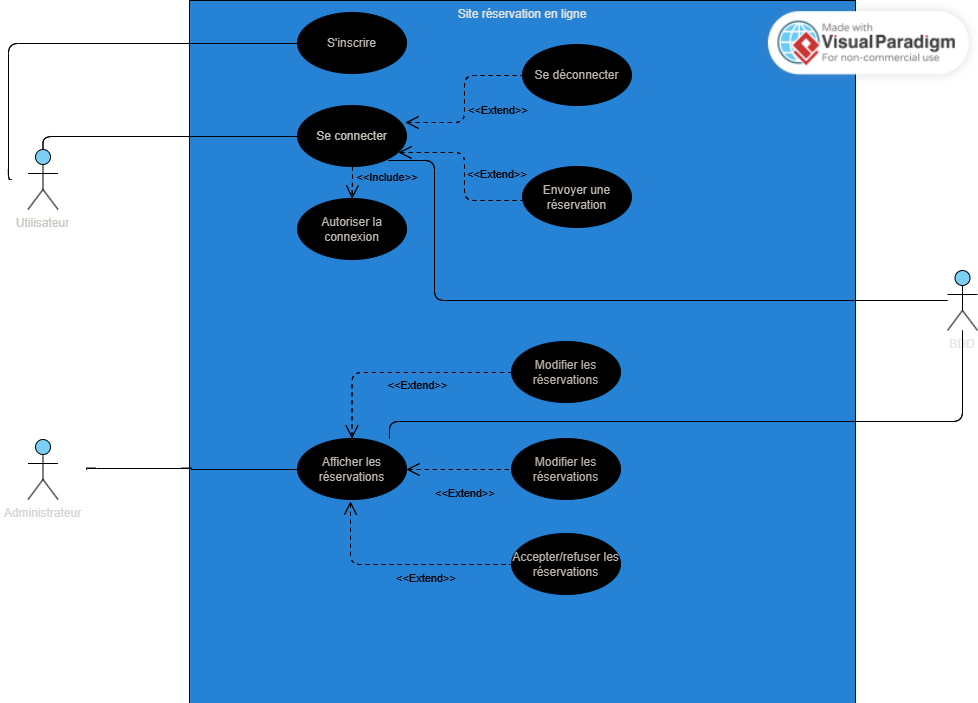
**Annexe 4 :**





**Annexe 5 :**

**Use case :**



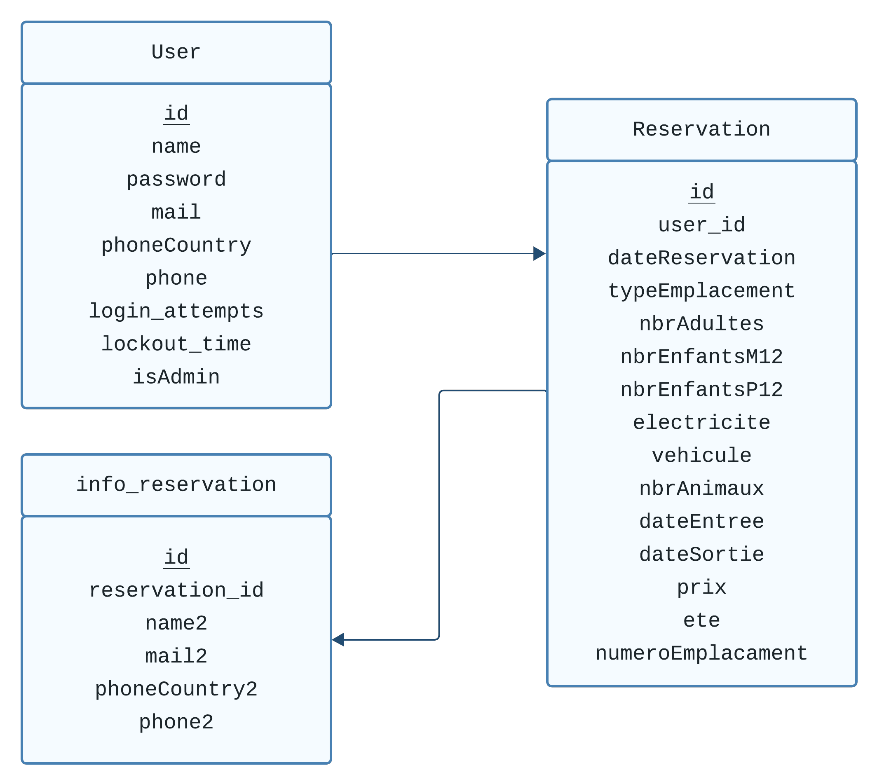
**Annexe 6 :**



**Annexe 7 :**



**Annexe 8 :**



**Diagramme de classes :**



**Annexe 9 :**



**Annexe 10 :**



**Annexe 11 :**



**Annexe 12 :**



**Annexe 13 :**

