

-Projet transverse développement application PHP/MySQL-

Sujet : Réaliser le site internet d'un camping

Sommaire :

- Conception de la Base de Donnée :
 - Équipe de Développement
 - Dictionnaire de Données
 - Définitions Fonctionnelles
 - Modélisation
 - Déploiement sur SQL Server
- Gestion du Projet:
 - Wireframe
 - Trello
- Page de Présentation Statique :
 - HTML
 - CSS
- Page d'Administration Dynamique
 - Équipe de Développement
 - Fichier de Fonctions modifiant la Base de Donnée
- Page de client dynamique:
 - Page de Réservation
 - Page "Mes Réservations"
 - Les Fonctions
- Conclusion

Conception de la Base de Donnée

Équipe de Développement

Lors de la conception de la Base de Donnée l'entièreté de la promotion EPSI B1 a travaillée ensemble sur le tableau blanc de la salle 11, sous la supervision de Monsieur Delcambre.

Dictionnaire de Données

Nous avons réalisé le dictionnaire de données en dessinant un à un les entités sur le tableau blanc puis pour chaque entités nous avons discuté ensemble de la nécessité de chaque données.

Chaque groupe a ensuite effectué leur propre mise en forme du dictionnaire sur leur documentation. Voici le notre :

Code	Entité	Type	Nature	Règle de calcul	Règle d'intégrité
id_client	Client	int	élémentaire	N/A	identifiant, auto_incre ment
nom	Client	text	élémentaire	N/A	non null
prenom	Client	text	élémentaire	N/A	non null
date_naissa nce	Client	date	élémentaire	N/A	non null
telephone	Client	string 50	élémentaire	N/A	
adresse	Client	text	élémentaire	N/A	
cp_ville	Client	string 50	élémentaire	N/A	non null
ville	Client	string 58	élémentaire	N/A	non null
mdp_client	Client	string 50	élémentaire	N/A	non null
mail	Client	string 50	élémentaire	N/A	non null, unique
id_admin	Administrateur	int	élémentaire	N/A	identifiant, auto_incre ment

login_admin	Administrateur	string 50	élémentaire	N/A	non null, unique
mdp_admin	Administrateur	string 30	élémentaire	N/A	non null
id_chalet	Chalet	int	élémentaire	N/A	identifiant, auto-incrément
id_type_chalet	Type_chalet	int	élémentaire	N/A	identifiant, auto-incrément
libelle	Type_chalet	string 50	élémentaire	N/A	non null
prix_base	Type_chalet	float	élémentaire	N/A	non null
nb_place	Type_chalet	int	élémentaire	N/A	non null
id_semaine	Semaine	int	élémentaire	N/A	identifiant, auto_incrément
numérp_semaine	Semaine	int	élémentaire	N/A	non null
date_debut	Semaine	date	élémentaire	N/A	non null
annee	Semaine	int	élémentaire	N/A	non null
date_fin	Semaine	date	élémentaire	N/A	non null
id_saison	Saison	int	élémentaire	N/A	identifiant, auto_incrément
type	Saison	string 50	élémentaire	N/A	non null
taux	Saison	float	élémentaire	N/A	non null
prix_modifie	Prix_special	float	élémentaire	N/A	
valide	Reservation	bool	élémentaire	N/A	
date_reservation	Reservation	date	élémentaire	N/A	non null
prix_total	Prix_special	float	calculé	(prix_base+prix_modifie)*taux	

Définitions Fonctionnelles

Les définitions fonctionnelles de premières règles ont été écrites séparément par chaque groupe depuis les entités sur le tableau blanc.

Dans un second temps, nous avons débattu ensemble oralement des définitions fonctionnelles de deuxièmes et troisièmes règles. Les voici par écrit :

Première Règle :

id_client → nom , prenom, date_naissance, telephone, adresse, cp_ville, mdp_client, mail

id_admin → login_admin, mdp_admin

id_type_chalet → libelle , prix_base, nb_place

id_semaine → numérp_semaine , date_debut, annee, date_fin ,

id_saison → type , taux

Deuxième Règle :

id_client, id_semaine, id_chalet → valide, date_reservation

id_chalet, id_semaine → prix_modifie

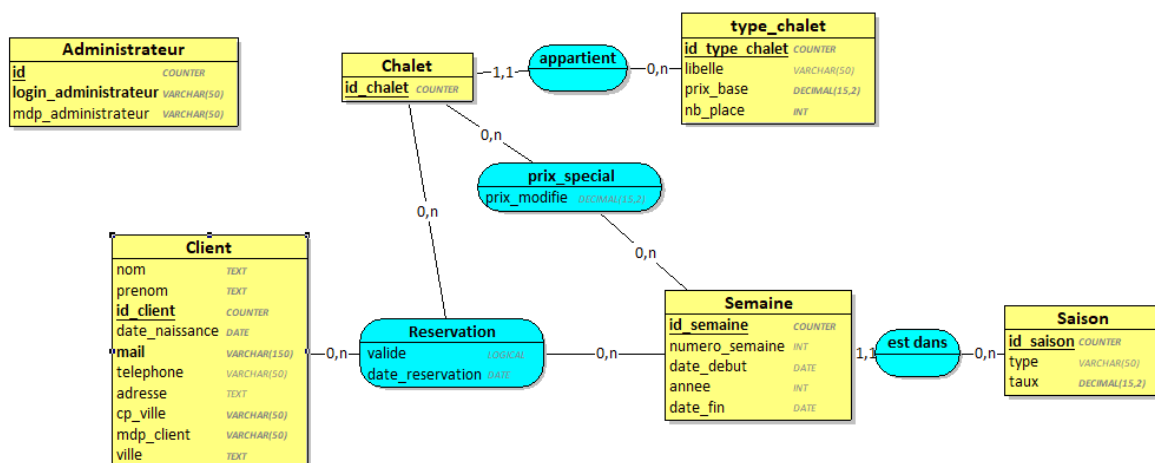
Troisième Règle :

id_chalet → id_type_chalet

id_semaine → id_saison

Modélisation

Le dernier événement de la séance de Brainstroming commune fut le dessin du modèle conceptuel de donnée sur le tableau blanc puis sa copie, par Gaëtan, sur le logiciel Looping et sa diffusion à l'ensemble des groupes. Le voici :



Grâce à Looping, nous avons à notre disposition un modèle linéaire de données, limité par les types de données fictifs interne à Looping :

```
Client = (id_client COUNTER, nom TEXT, prenom TEXT, date_naissance DATE, mail  
VARCHAR(150), telephone VARCHAR(50), adresse TEXT, cp_ville VARCHAR(50), mdp_client  
VARCHAR(50), ville TEXT);
```

```
type_chalet = (id_type_chalet COUNTER, libelle VARCHAR(50), prix_base DECIMAL(15,2), nb_place  
INT);
```

```
Saison = (id_saison COUNTER, type VARCHAR(50), taux DECIMAL(15,2));
```

```
Semaine = (id_semaine COUNTER, numero_semaine INT, date_debut DATE, annee INT, date_fin  
DATE, #id_saison);
```

```
Administrateur = (id COUNTER, login_administrateur VARCHAR(50), mdp_administrateur  
VARCHAR(50));
```

```
Chalet = (id_chalet COUNTER, #id_type_chalet);
```

```
Reservation = (#id_client, #id_chalet, #id_semaine, valide LOGICAL, date_reservation DATE);
```

Déploiement sur SQL Server

Notre responsable SQL, Gaëtan Tiberghien, a décidé l'utilisation du moteur de base de données SQLserver. Son rationnel fut sa préférence pour l'utilisation de Microsoft SQL Server Management studio comme outil de développement. Nous avons réalisé un script sql qui a pour vocation la création automatique des tables de la base de données. Puis nous nous le sommes partagé grâce à Google Drive. Le Voici :

```
CREATE TABLE Client(  
    id_client int IDENTITY,  
    nom TEXT NOT NULL,  
    prenom TEXT NOT NULL,  
    date_naissance DATE NOT NULL,  
    mail VARCHAR(150) NOT NULL,  
    telephone VARCHAR(50),  
    adresse TEXT,  
    cp_ville VARCHAR(50) NOT NULL,  
    mdp_client VARCHAR(30) NOT NULL,  
    ville VARCHAR(58) NOT NULL,  
    PRIMARY KEY(id_client),  
    UNIQUE(mail)  
);
```

```
CREATE TABLE Type_chalet(  
    id_type_chalet int IDENTITY,  
    libelle VARCHAR(50),  
    prix_base DECIMAL(15,2) NOT NULL,  
    nb_place int NOT NULL,  
    PRIMARY KEY(id_type_chalet)
```

);

```
CREATE TABLE Saison(  
    id_saison int IDENTITY,  
    type VARCHAR(50),  
    taux DECIMAL(15,2) NOT NULL,  
    PRIMARY KEY(id_saison)  
);
```

```
CREATE TABLE Semaine(  
    id_semaine int IDENTITY,  
    numero_semaine int,  
    date_debut DATE,  
    annee int,  
    date_fin DATE,  
    id_saison int NOT NULL,  
    PRIMARY KEY(id_semaine),  
    FOREIGN KEY(id_saison) REFERENCES Saison(id_saison)  
);
```

```
CREATE TABLE Administrateur(  
    id int IDENTITY,  
    login_administrateur VARCHAR(50) NOT NULL,  
    mdp_administrateur VARCHAR(30) NOT NULL,  
    PRIMARY KEY(id),  
    UNIQUE(login_administrateur)  
);
```

```
CREATE TABLE Chalet(  
    id_chalet int IDENTITY,  
    id_type_chalet int NOT NULL,  
    PRIMARY KEY(id_chalet),  
    FOREIGN KEY(id_type_chalet) REFERENCES type_chalet(id_type_chalet)  
);
```

```
CREATE TABLE Reservation(  
    id_client int,  
    id_chalet int,  
    id_semaine int,  
    valide bit,  
    date_reservation DATE,  
    PRIMARY KEY(id_client, id_chalet, id_semaine),  
    FOREIGN KEY(id_client) REFERENCES Client(id_client),  
    FOREIGN KEY(id_chalet) REFERENCES Chalet(id_chalet),  
    FOREIGN KEY(id_semaine) REFERENCES Semaine(id_semaine)  
);
```

```
CREATE TABLE Prix_special(  
    id_chalet int,  
    id_semaine int,  
    prix_modifie DECIMAL(15,2),  
    PRIMARY KEY(id_chalet, id_semaine),  
    FOREIGN KEY(id_chalet) REFERENCES Chalet(id_chalet),  
    FOREIGN KEY(id_semaine) REFERENCES Semaine(id_semaine)  
);
```

Gestion du Projet

Trello

Tout d'abord, nous avons créé une conduite de projet, en y notant ce qu'il fallait faire, ce qui est en train d'être fait, ce qui est fait et ce que l'on pourrait faire si tout ce qu'on a prévu est fait.

Puis, nous avons lié toutes les étapes avec les personnes qui y étaient associées. Au fur et à mesure du temps et des tâches accomplies, nous changeâmes les tâches pour les mettre d'actualité.

Wireframe

Ensuite, nous avons commencé à travailler sur le wireframe.

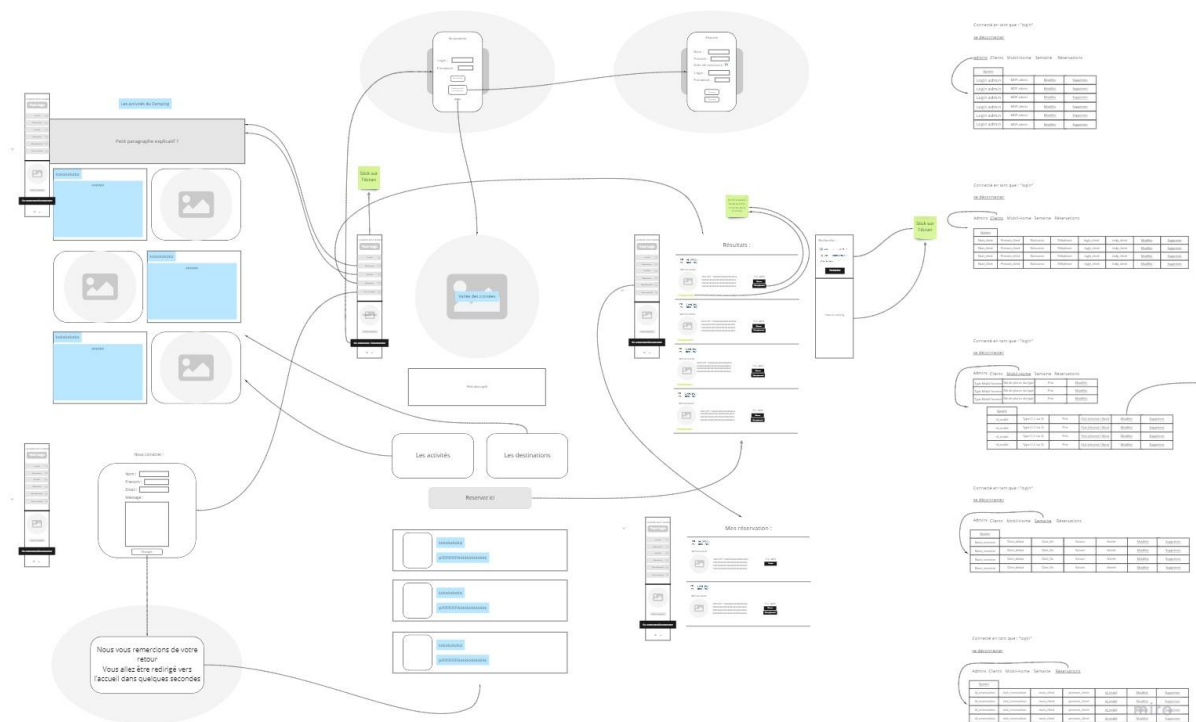
En 1^{er} lieu, nous avons mis en forme la partie cliente du site, en nous inspirant de certains sites de voyages sur internet. Nous avons alors décidé de faire un menu de navigation latéral fixé à l'écran. Nous avons ensuite convenus qu'il fallait créer les pages suivantes :

- Index : La page d'accueil du site, avec un carrousel, des liens vers d'autres pages, une présentation du camping et une partie news.
- Activités : La page activités présentes aux clients les différentes activités possibles dans le camping
- Destination : La page destination informe les clients des lieux touristiques proches du camping
- Réserver : Permet au client de chercher des mobil-homes disponibles en y indiquant la date de début de réservation, la date de fin et le nombre de personnes.
- Mes réservations : Cette page permet au client de voir les réservations qu'il a effectuées
- Nous contacter : affiche un formulaire pour que les clients puissent avoir un lien direct avec l'administration

Il y a également la possibilité de se connecter ou de créer un compte pour pouvoir réserver des mobil-home d'autant plus que le changement de langue est présent.

Dans un 2nd temps, nous avons réfléchi à la partie admin, où nous ne mettrons pas de CSS contrairement à la partie cliente, car celle ci n'est que fonctionnelle.

Dans cette partie administrateur, nous pourrons interagir avec les comptes admin, les clients, les différents mobil-homes et types de mobil-homes, les semaines et les différentes réservations des clients.



Page de présentation Statique

Cette partie a été effectuée par Aurélien Vanbelle , Aya et Rémi. Aurélien s'est occupé des pages html et Rémi et Aya du CSS.

Nous avons commencé par nous former à “Bootstrap” : c’est un framework utilisant les langages HTML, CSS et Javascript qui fournit aux développeurs des outils pour créer un site plus facilement tout en étant “responsive” à tout type d’écran.

HTML

A) La page d'accueil:

En premier temps, nous avons travaillé sur la page d'accueil. Dans cette page d'accueil, nous avons placé chaque élément dans une balise “DIV”, que ce soit pour le descriptif ou dans le sidebar, chaque “DIV” est constitué d’une classe pour aider la personne s’occupant de cela, la balise nous permet de mieux s’organiser dans les changements et aussi que la personne s’occupant du CSS puisse se retrouver à travers le code. Nous avons décidé de faire la navbar principalement en Bootstrap. Chaque gros éléments que se soit la description ou les news de la page web sont dans une “DIV”. Une slide de photo a été mise en premier plan de l'accueil du site pour que les clients se projettent et que cela leur donne envie d’en savoir plus.

B) Les pages de connection, inscription et de contact:

Dans un second temps, nous nous sommes occupés des pages de connexion, d’inscription et de contact. Pour ces pages, le code HTML sera mis dans une balise “FORM” qui va contenir des contrôles interactifs permettant à l'utilisateur d’envoyer des données au serveur web. Pour créer des champs que l'utilisateur va devoir remplir pour s’inscrire ou envoyer une demande au personne gérant le site nous allons utiliser les balises “LABEL” qui représente une légende pour un objet de l’interface et avec celui-ci, la balise “INPUT” pour que l'utilisateur puisse saisir des données. Pour chaque balise “INPUT”, nous avons mis un type, que ce soit pour un texte (text), la date (date), la mot de passe (password), un bouton envoyer (submit), et ensuite le code pour faire la relation entre la page connection et inscription avec le code “onclick=“window.location.href”. Chaque champ à remplir est obligatoire car nous avons mis le code “required” qui nous permet de faire cela. Chaque nouveau champs est intégré dans un “DIV”, l’explication de cette balise est au-dessus.

C) Les pages destinations et activités

Pour ces deux pages, qui sont similaire à la page d'accueil car nous avons repris le code de la navbar et nous avons aussi repris le même schéma que les autres pages, ce qui veut dire que nous avons mis à chaque balise de type "P" qui sert pour écrire, ou bien les balises de titre ou d'image, dans une balise "DIV". Pour la mise en page, des activités ou des destinations, l'enchaînement titre, image et texte est le mieux car au tout début, le titre donne une idée de l'activité proposée et ensuite, l'image et le texte qui sont tout autant importante pour sa présentation.

D) La navbar

Le lien et l'accès de chaque page du site se fait directement dans la navbar. Pour accéder de la page d'accueil à la page destination, par exemple, nous avons utilisé "href" dont on va mettre à l'intérieur le lien de nos pages. Par exemple, lorsque vous allez cliquer sur destination depuis la navbar, cela va vous envoyer directement sur la page concernée car ce bouton sera en lien avec la demande de l'utilisateur.

CSS

Nous réalisâmes la navbar qui apparaît sur différentes pages. Son design est clair et réfléchi pour permettre au client d'accéder aux pages pendant sa navigation plus facilement.

De ce fait, nous avons commencé à réaliser la conception des pages suivant le wireframe (expliquée auparavant) tout en ajoutant des couleurs neutres et reposantes qui apaise la vue.

L'accueil est la première page que voit l'utilisateur. Nous sommes parties sur des images du camping qui font rêver pour attirer le client et le pousser à réserver. Depuis le navbar, le client peut accéder à toutes les pages disponibles. Nous sommes parties sur des pages avec des designs qui se rapprochent pour être le plus simple et précis possible.

Nous avons décidé d'inclure du css dans le code html quand nous avons remarqué le besoin. Faire cela de cette façon nous à sembler plus clair pour identifier les différents besoins de chaque composants et nous aider quand nous avons rencontré des difficultés avec Bootstrap. Par exemple, nous avons inclus du style pour les marges et le rembourrage (padding) qui nous a facilité le travail pour différents composants durant le codage.

Page d'Administration Dynamique

Équipe de Développement

Afin de réaliser l'interface administratif du site, nous avons chargé Gaëtan Tiberghien de son développement. Sur la fin du projet, il a été assisté par Denis Jongmanee qui a effectué des corrections sur le CRUD des Chalets et a terminé la sécurisation de l'utilisation de la page.

Fichier de Fonctions modifiant la Base de Données

Dans un premier temps, un fichier fonction_sql.php fut écrit. Il regroupe les fonctions php effectuant des modifications sur le contenu de la base de données ainsi que quelques fonctions utilisées par la sécurité.

En entête se trouve un commentaire qui a servi de feuille de route durant l'écriture de ce fichier.

```
//Fonction sql : x fait et v idiotproof
/* Les tableaux interactions :
    list des administrateur
        ajout xv
        modification xv
        suppression x
    liste des chalets
        ajout xv
        suppression x
    liste des clients
        ajout xv
        modification xv
        suppression x
    liste des semaines
        ajout xv
        modification xv
        suppression x
    liste des reservation
        ajout xv
        suppression xv
    liste des prix_spéciaux par chalet par semaine
        ajout et modification xv
*/
```

Les fonctions utilisent une connexion PDO, issus du fichier connexionBDD.php à la racine du site. Elles effectuent des requêtes préparées.

CRUD des Administrateurs

Le développement du CRUD des administrateurs fut le plus lent. En effet, c'est lors de son écriture que l'architecture interne de la page fut inventée et testée.

Cette architecture se base sur une barre de navigation composée de lien vers chacun des CRUD, de deux niveaux de switch ainsi que de pages php intermédiaires de traitement.

La barre de navigation charge dans l'url le nom du CRUD qui est alors afficher.

Les boutons "Ajouter" et "Modifier" remplace le CRUD par un formulaire qui est envoyé à une page de traitement add_admin.php et modif_admin.php respectivement.

Suite à cela, le CRUD s'affiche de nouveau à l'écran.

Si le traitement est impossible alors il s'agit du formulaire qui est affiché, accompagné d'un message d'erreur.

Le bouton "Supprimer" lance la fonction suppr_admin() qui supprime l'administrateur afficher sur la même ligne.

CRUD des Clients

Le CRUD des Clients a été réalisé dans la continuité du CRUD des Administrateurs.

Il utilise les pages intermédiaires add_client.php et modif_client.php.

CRUD des Semaines

Le CRUD des Semaines a été réalisé dans la continuité du CRUD des Clients.

Il utilise les pages intermédiaires add_semaine.php et modif_semaine.php.

CRUD des Chalets

Le CRUD des Chalets a été réalisé dans la continuité du CRUD des Semaines.

Il utilise les pages intermédiaires add_chalet.php et modif_chalet.php.

Il a des particularités : la possibilité de voir l'état, la gestion des prix spéciaux du chalet par semaine et la possibilité de valider un chalet en attente de paiement sur son bouton "Modifier" et l'affichage des trois types de chalets disponibles. Il n'est pas

possible de modifier le prix spécial quand le chalet pour la semaine est en attente de paiement ou déjà réservé.

De plus, supprimer un chalet revient aussi à supprimer les réservations et les prix spéciaux du chalet pour éviter les erreurs de conflits des clés étrangères.

Ce CRUD a été corrigé par Denis durant les phases finales du projet.

CRUD des Réservations

Le CRUD des Réservations a été réalisé dans la continuité du CRUD des Chalets.

Il utilise les pages intermédiaires `add_reservation.php` et `modif_reservation.php`.

Sécurité d'Utilisation

Nous avons décidé dès le début du développement que nous ne voulions pas permettre d'erreur sql causé par l'utilisateur. Cette phase de sécurisation a été repoussée à la fin du Projet afin de nous assurer que l'ensemble des fonctionnalités soit disponible. Elle consiste en une vérification des données entrées et une obligation de remplir intégralement les formulaires de la page d'administration.

Page de client dynamique

Cette partie a été faite par Denis. La connexion à la base de données se fait par la bibliothèque PDO et utilise lorsqu'il est possible des requêtes préparées.

Page de Réservation

Cette page permet de voir l'état de l'ensemble des chalets et d'en réserver un ou plusieurs. Pour que le client puisse faire une réservation, il doit passer par le formulaire de recherche en précisant la date de début, la date de fin et le nombre de personnes. Le client ne peut pas saisir les dates antérieures au jour même car les chalets ne sont plus alors disponibles. En cliquant sur rechercher, il obtient l'ensemble des chalets qui répond aux conditions données et il pourra voir si les chalets sont libres ou déjà réservés. Il a accès à l'ensemble des informations des chalets notamment le prix du chalet.

Sécurisation : Toutes les fonctionnalités sont sécurisées pour que le client ne puisse pas avoir de message d'erreur lors de la saisie de données : la date de début avant la date de fin ou réserver un chalet qui l'est déjà. Cette page est accessible même étant déconnectée. Il faut cependant être connecté en tant que client pour pouvoir réserver un chalet.

Page “Mes Réservations”

Le client peut voir l'ensemble des réservations qu'il a faites, qui n'ont pas encore commencé et voir leur état, c'est-à-dire si la réservation est en attente ou confirmée. Il a aussi la possibilité d'annuler sa réservation s'il le souhaite.

Les Fonctions

Les requêtes sql et les fonctions qui peuvent être utilisées plusieurs fois se trouvent dans ce fichier. On y retrouve notamment la formule de calcul du prix total d'un chalet.

Conclusion

Nous avons conçu une application web fonctionnelle qui répond aux attentes du projet mais il existe de nombreuses améliorations possibles mais non travaillées par manque de temps (partie du site en anglais entièrement terminée, gestion de compte de l'utilisateur, cryptage des mots de passe, envoi de mails pour l'inscription et réservations de chalets, carte du camping, ajouter un type de chalet, compte instagram du camping, système de réduction ou offres...). Veuillez lire le fichier readme.txt pour accéder au site de façon efficace.