

# Le Parasol

Gautier Poursin, Andrew Farndon, Théo Lazzaroni, Joseph Elang

12 mai 2020

## 1 Introduction

L'objectif de ce rapport est de vous présenter notre site, **Le Parasol**, d'un point de vue technique. Tout d'abord, concernant la répartition des rôles. Théo s'est occupé de toute la partie CSS/HTML ainsi que des quelques animations présentes. Il a imaginé le visuel du site. Andrew s'est lui principalement occupé de la partie JavaScript. Gautier s'est occupé de la partie création de table SQL et de sa connexion au serveur puis du PHP avec Joseph.

Nous allons vous présenter dans ce rapport toutes les difficultés rencontrées et les solutions choisies pour les résoudre. Nous allons détailler les problèmes du site page par page. Il y a 5 grosses pages. Nous allons aussi vous présenter toutes les améliorations possibles à notre site ainsi que la structure de la base de données associée à notre site.



Figure 1: Logo du site

## 2 Structure de la base de données

La structure de la base de données du site se découpe en **6 tables** :

- Administrator(firstname,lastname,mail,passwd) : contient l'ensemble des administrateurs et leurs informations personnelles.

- Collaborator(firstname,lastname,mail,passwd) : contient l'ensemble des utilisateurs du site et leurs informations personnelles.
- Beach(name\_beach,characteristics,nudity,privacy,departement,localisation,frequentation,description,note) : contient l'ensemble des plages ainsi que les caractéristiques de celles-ci.
- City(name\_city,departement) : contient l'ensemble des villes où des plages sont situées et leurs départements.
- Commentaire(commentaire,plage,Note,nom,prenom,id) : contient l'ensemble des commentaires postés et les informations les concernant.
- Reponse(id, réponse,nom,prénom) : contient l'ensemble des réponses à des commentaires de même id.

### 3 Header/Footer

Avant de parler précisément de chaque page du site, nous allons vous présenter ici les 2 éléments communs à toutes les pages: **header** et **footer**. Le header est évolutif: si on est déconnecté, il y a une partie **Inscription/Connexion**. Sinon, cette partie se transforme en **Mon Compte**. Pour cela, Gautier a utilisé la technique **SESSION**. Lors de la connexion ou l'inscription, on lance **session\_start** en récupérant le nom et le prénom de l'utilisateur. Puis dans le header, si **SESSION['firstname']** n'est pas vide, alors on affiche **Mon Compte**.

Concernant le footer, ce dernier est similaire pour toutes les pages. La difficulté principale résidait dans la partie visuelle du footer. De manière générale, nous avons choisi de nous inspirer de designs épurés et assez minimaliste pour ne pas surcharger notre site de contenu et s'assurer d'aller à l'essentiel : trouver une plage qui convient à nos envies.

```
<?php if (isset($_SESSION['firstname'])) {
    echo '<a href="compte.php">Mon Compte</a>';
}
else {
    echo '<a href="registration.php">S inscrire/Se connecter</a>';
}
?>
```

Figure 2: Mon compte ou S'inscrire/Se connecter

### 4 Page d'accueil/Index

La page d'accueil est probablement la page la plus complexe du site . On peut y remarquer la carte de France, qui en est le point clé. Cette carte de

France est interactive. Un clic sur un département renvoie toutes les plages de ce département. Cette carte mixe donc Javascript, HTML, CSS et PHP. La balise **href** et la technique **GET** en PHP permettent de récupérer le nom du département puis de rentrer dans le formulaire de recherche de la page **Chercher une plage** ce nom de département et de lancer une recherche. De plus, Théo a ajouté une barre de recherche sur le côté pour rechercher les plages d'une ville en particulier. Là aussi, on est renvoyé sur la page de recherche avec le formulaire complété par le nom de la ville via la méthode **POST**.

Ensuite, la page d'accueil possède un bouton qui permet de se connecter. Ce bouton renvoie à la page **Inscription/Connexion**. Si jamais on est déjà connecté, alors cette partie se transforme en case qui recense le nom, le prénom ainsi que le statut de la personne (si c'est un simple utilisateur ou si c'est un administrateur). Pour savoir si la personne est connectée ou non, la fonction **isset** en PHP vérifie si **SESSION['firstname']** est vide ou non. Si c'est le cas, on affiche le bouton **Se connecter**.

```
<?php if (!isset($_SESSION['firstname'])) {?>
<div class="w3-col w3-animate-opacity" style="width:30%" >
<button class="w3-button w3-block w3-black" onclick="location.href='registration.php'">Se connecter<
</div>
```

Figure 3: Bouton Se connecter

De plus, Théo a rajouté un sous-menu à gauche de la carte. Il mixe PHP, HTML et JS. Le JS permet lors du clic sur l'icone d'afficher le menu. Le PHP permet d'afficher soit **Mon Compte**, soit **Se connecter** selon l'état de la personne naviguant sur le site.

## 5 Chercher une Plage

Cette partie permet aux utilisateurs de rechercher les plages dans toutes la France. C'est pour cela qu'un simple formulaire fait l'affaire. Cependant, le principal problème réside dans le fait qu'il faut rentrer un mot exact dans le formulaire. Par exemple, si on rentre "Sable" au lieu de "sable", alors la recherche renverra "Aucune plage", alors qu'il y a des plages correpondantes. Pour remédier à cela, nous avons pensé à faire une autocomplétion avec du JavaScript. Cependant, la partie CSS de l'autocomplétion n'est pas terminée, le menu deroulant des propositions n'est pas opaque, c'est une erreur que nous n'avons pas réussi à résoudre jusque ici.

Théo a donc prérempli les formulaires avec les mots exact qu'il faut indiquer: "sable/galets", "nudiste/non nudiste"... La seconde solution consiste à rentrer dans la base de données toutes les différentes écritures possibles:

"Sable", "sable", "SABLE", etc. Cependant, cette méthode risquerait de surcharger la base de données et cela pourrait ralentir les recherches. Lorsqu'un mot est orthographié différemment que dans la base de données, il est indiqué à l'utilisateur que ce mot ne correspond pas aux attentes lorsqu'il clique sur **Rechercher**. C'est pour cela que nous avons utilisé du JavaScript afin d'avoir une autocomplétion, Andrew s'en est chargé. Nous avons ici encore un problème avec l'opacité du menu déroulant.

Pour rechercher une plage, il n'est pas nécessaire de remplir tout le formulaire. On peut par exemple rechercher les plages d'un département, d'une ville sans se soucier des autres caractéristiques. Gautier a donc effectué différents cas **if** dans le code PHP. Il n'a pas trouvé d'autre méthode que d'énumérer 1 à 1 les différents cas, il y en a donc 6! soit 720 cas possibles. Pour simplifier l'écriture du code, on a décidé de se passer de certains cas et de prendre seulement en compte les cas utiles. Par exemple, on peut rechercher une plage nudiste dans une ville ou un département précis. En revanche, rechercher toutes les plages publiques de France n'a pas vraiment d'intérêt donc cela n'a pas été codé.

Les résultats de la recherche sont donnés sous forme de liste. On indique tout d'abord le nom puis la description de la plage et puis on informe les caractéristiques, la ville associée etc. A l'avenir, si nous devons continuer ce site, nous essaierions d'améliorer cette page de résultat. En effet, on souhaiterait qu'il soit possible de cliquer sur le nom d'une plage pour nous redirigez vers une nouvelle page, avec une description et des informations supplémentaires. Par exemple, on aurait en plus les commentaires associés, les avis ou des informations pratiques comme sa localisation. Gautier a tenté de faire ça rapidement avec un bouton "Voir +" à côté du nom de la plage. Cependant, il n'a pas réussi à afficher les informations et récupérer les commentaires associés seulement à cette plage. Une méthode consisterait, pour chaque plage, à créer une nouvelle page avec une recherche SQL associée plage. Il aurait fallu le faire pour chaque plage et donc à chaque ajout de nouvelles plages. Cette méthode n'était pas optimale et donc écartée très rapidement. Les plages sont classées par ordre décroissant de note. Les notes étant attribuées par les utilisateurs, on fait la moyenne de chaque note. De plus, en dessous de chaque plage est affiché l'ensemble des commentaires associés. chaque commentaire étant posté dans la section "Parler d'une plage".

Une deuxième version de l'affichage a été proposée en supposant que nous pourrions permettre aux utilisateurs d'uploader une image de la plage et ainsi l'afficher lorsque l'on recherche une plage, en y ajoutant un affichage dynamique réalisé en CSS et affichant sobrement les caractéristiques de la plage puis les commentaires.

## 6 Parler d'une plage

Cette page permet aux utilisateurs d'interagir entre eux. Par exemple, si un utilisateur poste un commentaire sur une plage, alors il est possible de lui répondre.

Afin de poster un commentaire, il faut avant tout pouvoir être connecté. Un code PHP permet soit d'afficher un bouton **Poster un commentaire**, soit un bouton **Se connecter**. La vérification se fait avec la méthode `isset`, on vérifie si `$_SESSION['firstname']` n'est pas vide.

Lorsqu'on est connecté, il est en plus possible de répondre à un commentaire. Cette réponse s'affiche juste en dessous du commentaire correspondant. Si nous ne sommes pas connecté, il n'y a pas de bouton répondre: il faut se connecter au préalable. Nous n'avons pas prévu d'avoir des tables "Réponses" et "Commentaires" dans la base de données. Mais il était nécessaire de les ajouter. En effet, chaque commentaire et chaque réponse doivent être gardés en mémoire. De plus, un autre problème est apparu : comment garder en mémoire le commentaire auquel on répond ? Pour cela, nous avons décidé de donner à chaque commentaire un ID unique. Cet ID sera affiché entre parenthèses à la fin de chaque commentaire. Ainsi, lorsqu'un utilisateur veut répondre, il doit informer l'ID du commentaire auquel il répond. Cette technique est simple, mais elle n'est pas efficace ni même pratique. En effet, il faut garder en mémoire le numéro du commentaire. Une amélioration serait d'avoir automatiquement le numéro de l'ID. L'autre inconvénient est la présence de l'ID entre parenthèse. Sur aucun autre site on ne voit ça. L'ID est gardé automatiquement en mémoire et on affiche jamais cet ID.



Figure 4: Affichage lorsqu'on est déconnecté



Figure 5: Affichage lorsqu'on est connecté avec réponse

Cette page diffère aussi lorsqu'on est administrateur. En effet, un administrateur peut supprimer des commentaires qu'il juge inappropriés. Pour cela, il lui suffit de cliquer sur la croix rouge se situant au dessus du commentaire. Il faut qu'il indique, comme précédemment, le numéro du commentaire qu'il souhaite supprimer. La suppression d'un commentaire entraîne automatiquement la suppression de toutes les réponses associée à l'aide de la fonction SQL **DELETE**. Un administrateur peut lui aussi poster un commentaire et répondre à d'autres commentaires. Les utilisateurs ne sauront pas qu'il s'agit d'un administrateur qui leur répond.

Lors de l'ajout d'un commentaire, un utilisateur doit mettre une note. Cette note est enregistrée dans la table commentaire. Ensuite, ces notes sont réutilisées pour calculer la note moyenne d'une plage. Cela est un point clé de notre site, puisqu'il s'agit de répertorier et de noter les plages. Pour calculer la moyenne, Gautier a fait appel à 2 requêtes SQL: une **SELECT** pour prendre toutes les notes, une **UPDATE** pour mettre à jour la moyenne dans la table Beach.

## 7 S'inscrire/Se connecter

Dans cette partie, un utilisateur peut s'inscrire ou se connecter au site. Andrew a mis en place un formulaire javascript pour indiquer à l'utilisateur lors de son inscription qu'il manque des champs. D'ailleurs, chaque champ non rempli devient rouge. De même, lorsque 2 mots de passe non identiques sont rentrés, alors l'utilisateur est averti lorsqu'il clique sur "S'inscrire". Si jamais l'utilisateur rentre une adresse mail non valable ou une adresse mail déjà utilisée, il sera averti après avoir cliqué sur "S'inscrire", puisque nous n'avons pas réussi à accéder à la base de donnée directement avec du JavaScript. Nous avons donc codé une vérification qui regarde si ce mail est déjà utilisé, via une commande **SELECT**. Pour la connexion, il n'y a pas de JavaScript dessus. On va directement chercher dans la base de donnée les informations.

Lorsqu'on est administrateur, il suffit de cliquer sur le bouton administrateur pour accéder à la page de connexion. Ce qui diffère de la page précédente, c'est que l'on va chercher/ajouter des informations dans la table "Administrator". Gautier n'a pas trouvé de solution pour faire autrement. Il doit y avoir probablement des méthodes plus sécurisée, puisque la seule sécurité réside dans le fait qu'il faut avoir le mot de passe administrateur pour s'inscrire en tant qu'administrateur.

Un utilisateur peut être à la fois administrateur et utilisateur simple. c'est à dire qu'avec une adresse mail, on peut avoir 2 compte et c'est le seul cas possible.

## 8 Mon Compte

Cette section varie si l'on est utilisateur ou administrateur. Un administrateur à plus de droit: il peut ajouter/ modifier une plage, supprimer un utilisateur/commentaire. C'est sur cette page que l'on peut retrouver le **CRUD**. Il suffit de cliquer sur supprimer un utilisateur et la liste de tous les utilisateurs apparait. Pour en supprimer, il suffit de rentrer son adresse mail ainsi que le mot de passe administrateur.

Il est aussi possible de modifier son mot de passe et cela est valable pour tous les utilisateurs et administrateurs. On met directement à jour la base de donnée via une fonction SQL **UPDATE**.

Un utilisateur a accès à son historique. Sur cette page sont recensés l'ensemble de ses commentaires postés ainsi que ses réponses. S'il s'agit d'un commentaire, alors l'utilisateur peut aussi voir la note qu'il a donnée et la plage correspondante. S'il s'agit d'une réponse, l'utilisateur voit seulement la réponse qu'il a posté. Enfin, l'utilisateur voit exactement son nombre de commentaires postés ainsi que son nombre de commentaires postés.

Pour améliorer cette page, on pourrait modifier l'affichage en affichant les meilleurs réponses à ses commentaires ou bien le commentaire associé à ses réponses à lui. De plus, pour afficher l'historique, on va chercher dans la table "Commentaire" et "Reponse" les commentaires et réponses correspondant au nom et prénom de l'utilisateur. Cependant, il est possible que 2 utilisateurs aient les mêmes nom et prénom. Il faudrait modifier le code et la BDD en allant prendre les commentaires et réponses associés à l'adresse mail, puisque celle-ci est unique.

Nous avons pensé à ajouter un système de photo de profil permettant de distinguer un peu plus les comptes et rendre l'expérience utilisateur plus agréable. Cependant, comme pour les photos des plages, nous ne sommes pas parvenus à mettre en place un système de mise en mémoire de photos.

## 9 Contact

Cette section est une page que nous n'avons pas exploitée. En effet, cette page serait intéressante si nous avions la possibilité d'envoyer des mails avec ce site. Or, Gautier n'a pas réussi à mettre en place l'envoi de mail en local. C'est l'une des principales améliorations à faire : réussir l'envoi de mail. Cela serait utile par exemple lors de l'inscription de l'utilisateur: il recevrait un mail de bienvenue. Il pourrait aussi recevoir un mail lorsqu'il a oublié son mot de passe. C'est d'ailleurs pourquoi nous n'avons mis en place ce système de "Mot de passe oublié". Il est possible de changer son mot de passe lorsqu'on est connecté mais pas lorsqu'on est déconnecté. L'envoi de mail peut se faire avec la fonction

`mail()` de PHP, mais il ne faut pas travailler en local.

Cette page est donc inutilisable ici, on ne peut pas interagir avec mais elle reste importante pour un site web.

## 10 Le futur de Le Parasol

Nous avons pensé à de nombreuses fonctionnalités à programmer si nous avions à continuer le projet dans la durée. Par exemple, l'expérience entre les utilisateurs reste simplement au stade d'échange de commentaires et de notes sur les pages. Nous avons pensé à ajouter un système de partage sur les réseaux sociaux ou même ajouter un forum pour que les utilisateurs partagent leurs bons plans etc. De même, le site paraît pour l'instant un peu vide, mais c'est parce qu'il se base entièrement sur la communauté qu'il va créer.

En outre, la sécurité du site n'a pas été testée puisqu'il est privé pour l'instant. Gautier a néanmoins crypté les mots de passe qui sont dans la base de données pour éviter les fuites mais nous n'avons pas vérifié la sécurité interne du script, par exemple si on essaie d'attaquer le site en envoyant des formulaires frauduleux.

## 11 Conclusion

Pour conclure sur ce projet, nous avons réalisé nos objectifs fixés initialement qui était de rendre un site web fonctionnel. La plupart des fonctionnalités souhaitées sont présentes. Nous avons vu aussi qu'il reste encore des choses à améliorer. Par exemple, il serait intéressant d'intégrer à la base de données des photographies de chaque page ou même d'envoyer des mails. Des améliorations peuvent aussi être possible en JavaScript, comme par exemple une meilleure visibilité des propositions d'autocomplétation.

Au niveau du code, celui ci peut être amélioré d'un point de vue visuel. Nous n'avons pas séparé les fonctions PHP des fichiers HTML. Habituellement, PHP et HTML sont bien distincts et les fichiers PHP sont inclus dans les fichiers HTML. Dans notre projet, ils sont mélangés dans un seul fichier. Nous pouvons améliorer aussi notre code sur le nommage des fichiers, certains noms ne sont pas assez clairs.