

# Rapport de projet Web

« *Move IT* »

ANTEUNIS Charles & GUITTON Thomas

## Table des matières

1. Objectif du projet Web.....	3
2. Description de notre projet : « Move IT ».....	3
3. Fonctionnalités souhaitées.....	3
3.1 Une carte personnalisée.....	3
3.2 Un compte utilisateur.....	3
3.3 Un référencement des spots.....	4
3.4 Un référencement des figures.....	4
3.5 Une fonctionnalité de suivi.....	4
4. Répartition des tâches.....	4
5. La base de données.....	5
6. La liaison de la base de données au site.....	5
7. La gestion d'utilisateurs.....	6
7.1 Les formulaires d'inscription / connexion / modification.....	6
7.2 La session utilisateur.....	6
8. Le design du site.....	6
9. La carte personnalisée.....	7
10. Conclusion.....	7

# 1. Objectif du projet Web

Ce projet a pour objectif de nous familiariser avec les outils de création d'un site web ainsi que les différents enjeux qui interviennent lors de la réalisation d'un projet concret. Cela passera par l'assimilation des langages tels que PHP, HTML et CSS, ainsi que la structure générale d'une application Web.

## 2. Description de notre projet : « Move IT »

L'idée du projet vient d'une passion commune pour un sport urbain : le Parkour. Du fait du manque de structures adéquates pour pratiquer ce sport, il est souvent pratiqué dans les rues en utilisant l'environnement urbain. La première difficulté est donc de trouver un endroit pour s'entraîner : on appelle cela un « spot ».

Nous avons donc pensé à une plateforme qui permettrait aux utilisateurs de partager entre eux leurs meilleurs spots d'entraînement. Le but est de faciliter la recherche de spots intéressants, proches des utilisateurs (géographiquement) avec une liste des différentes activités (figures) qu'il est possible de faire.

## 3. Fonctionnalités souhaitées

### 3.1 Une carte personnalisée

Pour faciliter la recherche de spots, nous souhaitons implémenter une carte donnant l'emplacement et le nom des spots les plus proches d'une ville que l'utilisateur a cherché.

### 3.2 Un compte utilisateur

Chaque utilisateur pourra disposer d'un compte modifiable qui lui permet de renseigner des informations comme sa ville de résidence, ses

années d'expériences, son âge... Un utilisateur connecté pourra alors ajouter un spot qu'il a découvert pour le partager avec les autres utilisateurs.

### 3.3 Un référencement des spots

Nous voulons que les spots puissent être notés pour être classés, et qu'une liste des différentes figures praticables sur ce spot soit accessible aux utilisateurs. Un administrateur devra cependant pouvoir supprimer un spot pour éviter les spots inexistantes ou non appropriés.

### 3.4 Un référencement des figures

Les utilisateurs pourraient aussi rechercher des spots en fonction des figures qui y sont praticables. Chaque figure se verrait attribuer une note de difficulté ce qui permettrait d'affiner les critères en fonction des années de pratiques de l'utilisateur.

### 3.5 Une fonctionnalité de suivi

Nous voulons qu'un utilisateur puisse suivre un spot pour être informé de son activité : lorsque quelqu'un va s'y entraîner par exemple. Un utilisateur pourrait aussi suivre un autre utilisateur (avec son accord) pour pouvoir être informé lorsque son ami décide d'aller s'entraîner à tel spot.

## 4. Répartition des tâches

Thomas GUITTON est en charge : du schéma de la base de données, de l'implémentation SQL de la base de données, de l'apparence de la page d'accueil, de l'implémentation de la carte personnalisée et des fonctionnalités qui lui sont attachées.

Charles ANTEUNIS est en charge : de l'implémentation des fonctions de lien entre la base de données et les pages PHP, de l'inscription / la connexion / la gestion d'utilisateurs, de l'ajout de spot par un utilisateur, des liens entre les différentes pages et du design global du site.

Nous avons travaillé avec un git pour une gestion plus efficace du code.

## 5. La base de données

Nous avons implémenté 4 tables au total :

- une table « user » pour gérer les utilisateurs et leurs informations
- une table « spot » gérant le nom, la ville, l'emplacement et la note
- une table « move » pour lister les figures et leurs difficultés
- une table de jointure « spotXmove » pour obtenir les différentes figures praticables à un spot, ou les spots où l'on peut pratiquer telle figure.

Cette table se construit avec pour clé primaire un couple de clés étrangères correspondant aux clés primaires des tables user et spot.

## 6. La liaison de la base de données au site

Pour chaque table de la BDD(base de données), une classe PHP Table et une classe TableRepository sont implémentées. La classe Table est une représentation sous forme d'objet de la table SQL, avec des getters et setters. Cela permettra d'accéder facilement à tous les attributs de chaque élément provenant d'une table, via `$element->getAttribute()`. La classe TableRepository concentre les fonctions permettant d'effectuer des requêtes SQL sur la table en question. Ainsi, tous les accès à la BDD au sein des pages se fera via des fonctions de TableRepository.

Les requêtes sont exécutées à l'aide de la fonction `query()` ou des fonctions `prepare()` puis `exec()` qui permettent de se prémunir contre l'injection SQL. Des difficultés de syntaxes dans les chaînes de caractères des requêtes ont été rencontrées mais résolues notamment avec la fonction `$this->connection->quote($var)`.

Les classes de la table de jointure « spotXmove » ne fonctionnent pas à cause de conflits d'inclusion : il faut faire appel à des fonctions de SpotRepository ainsi que MoveRepository et cela ne fonctionnait pas.

## 7. La gestion d'utilisateurs

### 7.1 Les formulaires d'inscription / connexion / modification

L'inscription et la connexion se font au moyen de formulaires HTML. Lors de la saisie par l'utilisateur, le formulaire vérifie si les données sont conformes : adresse mail valide, mot de passe obligatoire, années d'expérience positives et entières etc. La vérification se fait à l'aide des attributs HTML comme `required=true` ou `type=number min=0`. Une seconde vérification est effectuée dans les fonctions `TableRepository` dans le cas où certains attributs sont vides par exemple.

Le principe est le même pour l'ajout de spot lorsque l'utilisateur est connecté.

### 7.2 La session utilisateur

Une fois l'utilisateur connecté, il doit pouvoir naviguer à travers les pages sans devoir se re-connecter à chaque fois. Cela est fait à l'aide de la fonction `session_start()` et de la variable `$_SESSION['mail']` qui contient le mail de l'utilisateur actuel. La session est commune aux différentes pages. Les informations de l'utilisateur se récupèrent avec une fonction de `UserRepository` (`fetchOneByMail($mail)`) grâce au mail. La déconnexion se fait via un `unset($_SESSION['mail'])`.

On peut vérifier si un utilisateur est connecté grâce à la fonction `isset($_SESSION['mail'])`.

## 8. Le design du site

Le design du site est fait avec une feuille CSS à part. Les différents éléments des pages disposent d'attributs `class` et `id`, ce qui assure une harmonisation du design. Des attributs comme `:hover` sont aussi utilisés pour afficher des menus déroulant ou des changements de couleurs au passage de la souris. La majorité des éléments dans les pages sont disposés

dans des `flexbox` qui permettent d'adapter le design à plusieurs types d'écrans, de largeur, de hauteur...

Les parties importantes des pages comme la navigation, la bannière ou le footer sont implémentées avec des fonctions PHP pour plus de commodité à les appliquer à chaque page et assurer une harmonie.

## 9. La carte personnalisée

La carte est réalisée à l'aide de Leaflet, une librairie JavaScript open source pour les cartes interactives, utilisée par OpenStreetMap. Les fonctions JavaScript de créations de la carte au moyen de l'API sont placées dans des balises `<script>` de la page d'accueil.

Il est possible de zoomer et placer la carte à un endroit souhaité avec `L.map('mapid').setView([latitude, longitude], zoom)`, mais on ne récupère pas encore la latitude et longitude à partir de l'entrée d'une ville par l'utilisateur. On parcourt ensuite tous les spots de la BDD avec une fonction de `SpotRepository`, et on ajoute chaque spot à la carte avec `L.marker(latitude, longitude).addTo(myMap)`. On ajoute ensuite un pop-up de sorte à ce que le nom du spot soit affiché lorsque l'utilisateur clique sur la carte : `marker.bindPopup(nomDuSpot)`.

## 10. Conclusion

L'objectif était de réaliser une plateforme de recherche de spots de Parkour avec une carte et une système de suivi entre utilisateurs et entre spots et utilisateurs. Nous n'avons pas implémenté toutes les fonctionnalités souhaitées, toutefois une partie de la plateforme est tout à fait fonctionnelle : les spots sont affichés sur la carte, les utilisateurs peuvent ajouter des spots, la gestion des utilisateurs est complète.

Ce projet nous a permis d'avoir une idée de comment concrétiser un projet dans le domaine du web et nous a apporté certaines compétences

techniques tout en nous poussant à être autonomes et à apprendre par nous-mêmes.