



DEVELOPPEMENT WEB - Année 2018/2019

CLASSE PREPARATOIRE CPI2G2

PROJET SEMESTRE 2 :

**KUNU**

---

Made by :  
Lilian Naretto, Romain Séré,  
Nathan Serfaty, Ryan Filali

# Table des matières

0.1	Structure du code . . . . .	2
0.1.1	Côté client . . . . .	2
0.1.2	Côté serveur . . . . .	4
0.1.3	Projet KUNU améliorations . . . . .	4
0.2	Choix de mise en forme . . . . .	5
0.2.1	Les difficultés rencontrées . . . . .	5
0.3	Répartition du travail . . . . .	6
0.3.1	Répartition simplifiée . . . . .	6
0.4	Bilan . . . . .	7

## 0.1 Structure du code

Le code du projet **KUNU** est divisé en différents fichiers avec différents langages que l'on peut regrouper en deux parties principales :

- *Côté client* : Cette partie regroupe les fichiers HTML ,CSS et JavaScript. Ils permettent d'insérer le contenu du fichier que le client verra et mettre en page ce contenu de sorte qu'il attire et soit facile à comprendre.
- *Côté serveur* : Cette partie regroupe les codes PHP et SQL. Ils permettent de communiquer avec les bases de données du serveur, créer du html et changer le contenu de la page actuelle.
- *Projet KUNU* : L'ensemble des fichiers sont liés les uns aux autres ; ce qui permet de surfer sur le site d'une page à l'autre. Le code HTML est inclus dans les fichiers php alors que le CSS et Javascript sont sur des fichiers à part.

Dans cette partie du rapport, nous allons explorer les différentes fonctions du site codées durant ce projet.

### 0.1.1 Côté client

Cette partie correspond aux codes HTML, CSS et Javascript . Nous avons créé autant de fichiers HTML que de pages sur le site. Le code étant couplé à du PHP, les fichiers sont en php . Chaque fichier php est relié à un fichier css qui permet d'organiser le contenu des pages. Pour faciliter notre travail, nous avons utilisé Bootstrap et JQuery qui sont étroitement liés.

- *connexion* : La page connexion est la première page du site. Cette page est reliée au fichier register.php. Elle ne comprend que du code html lié à du code CSS Bootstrap et nous permet de nous rediriger vers un espace pour se connecter ou s'inscrire. Cette nouvelle page s'appelle register2.php. Elle contient du code html pour créer le formulaire et du bootstrap pour la mise en page ainsi que les fichiers register1 et respectivement register2.css. Sans être connecté à un compte préalablement créé, on ne peut pas accéder aux autres parties du site. Nous serons donc directement redirigés vers la page register.php .
- *espace-client* : Une fois connecté, on arrive sur une page d'accueil client index.php qui nous fait une petite présentation des possibilités du site (activités,espace-client,boutique). Comme chaque page du site, elle comprend une partie header accouplée au code css bootstrap et une partie footer. Pour la mise en forme de cette page, une page index.css a été créée.
- *activité* : La page activité est une page d'accueil des activités qui comprend un bouton calendrier et carte. Tout est compris dans le fichier activite.php. Elle contient comme toutes les autres pages un header et un footer. La mise en page est gérée par activite.css et annexe-activite.css. Nous avons créé 4 pages activité, chacune représente une activité, chacune liée à une carte créée dans les fichiers respectifs : "cartel 1 à 4".js. Nous allons

maintenant séparer activité en 2 parties : calendrier et carte.

- *calendrier* : Le calendrier est codé dans la page `calendrier.php` . Ce fichier est lié contient du JS , `fullcalendar.js`. Il a été réalisé par Adam Shaw. Nous avons récupéré son code et apporté des modifications. Son code est très clair et permet d'apporter d'énormes modifications avec les 244 premières lignes de code mais la suite est un ensemble de fonctions assez compliquées.  
Le fichier `calendrier.php` contient du code js dont une partie qui est inutilisée mais elle permet de rajouter des activités sur le calendrier directement en cliquant dessus. On a souhaité bloquer cette fonction pour éviter que tout le monde puisse rajouter des activités sur le calendrier. Elle contient aussi la partie Navigation (header) et la partie footer qui est aussi sur toute les pages .php et du code css. Ce fichier est aussi lié à deux autres fichier CSS , `fullcalendar.css` et `fullcalendar.print.css` .
- *carte* : La carte est une map OpenStreetMap. On a fait ce choix car c'est open sources. Elle est codée sur un fichier `carte.js` où pour placer des marquages d'activités il suffit de mettre les coordonnées latitude et longitude dans le Js. Le fichier `carte.php` comprend en plus un tableau récapitulatif des activités avec un lien permettant d'accéder directement à d'autres pages `activité1.php` et respectivement 2 3 4 , permettant de voir une description précise de chaque activité avec une mini carte codée dans les fichiers `carte1` à 4 .js . On avait d'abord créé une map google map avec des calques grâce à my maps. Pour l'utiliser, il suffit d'intégrer cette ligne de code dans un fichier html : `<iframe src="https://www.google.com/maps/d/u/1/embed?mid=1Ywq41PthsCcKh4ZJAjjNhxJbJe4IH0hn" width="640" height="480"></iframe>` mais cela ne nous avait pas plus car il fallait payer pour avoir des fonctionnalités en plus et il fallait être connecté à un compte google pour créer la carte. On a donc associé cette carte au compte google de Nathan. Cependant, nous l'avons gardé dans l'idée où le client souhaiterait payer pour accéder à ses fonctionnalités.
- *boutique* : La boutique est codée dans le fichier `shop.php`. Du bootstrap y est utilisé grâce à `div class="container"` . Elle est mise en page grâce à un fichier `shop.css`. Chaque élément de la boutique a une page personnalisée avec plus d'info sur l'article qui peut ensuite être mis dans le panier pour être acheté.

### 0.1.2 Côté serveur

Cette partie correspond aux différents codes en langages PHP et SQL :

- *"serveur.sql"* : C'est un fichier qui permet de créer les différentes tables : products et user. Dans le cas où l'ordinateur qui essaye de se connecter au réseau de l'EISTI n'y arrive pas, il faut alors créer les bases de données manuellement grâce à ce dossier.
- *"tous les fichiers .php"* : Dans tous les fichiers, à part les deux premiers(register.php), il y a un petit morceau de code tout en haut permettant de vérifier que l'utilisateur est bien connecté afin d'afficher la page. Le fonctionnement du magasin se fait grâce à des fonctions PHP ainsi que la connexion et les différents changements d'information de compte.
- *"connexion"* : les fichiers server et server2.php permettent de vérifier la bonne connexion au compte ou une bonne création de compte c'est à dire ne pas avoir le même user que quelqu'un autre ou une mauvaise adresse mail. Les fichiers server3 et server4.php servent à vérifier les bonnes informations et changer de mot de passe ou de user dans la base de données. Ces fichiers s'appliquent à l'endroit où l'on peut changer les paramètres de compte.

### 0.1.3 Projet KUNU améliorations

Après avoir implémenté les différentes fonctionnalités disponibles sur notre site, nous nous sommes rendu compte que certaines ne fonctionnaient pas, ne s'arrangeaient pas correctement ou n'étaient pas très réalisables. Nous rassemblons donc dans cette partie, les fonctionnalités que nous aurions aimé voir fonctionner ou bien être implémentées sur notre site :

- *"compte administrateur"* : le compte administrateur permettant de gérer le site n'a malheureusement pu être intégré car nous ne savions pas comment faire pour différencier les droits "utilisateur" des droits "administrateur".
- *"système d'abonnements et historique d'utilisation"* : un système d'abonnement pour recevoir des magazines en lien avec le naturisme aurait été une fonctionnalité intéressante. Nous avons créé toute la mise en page de ce système mais ne sommes pas parvenus à enregistrer les abonnements choisis. Il en est de même pour ce qui est de l'historique d'utilisation.
- *"nombres d'items dans le panier"* : lors de la création du panier et du système d'achat dans le magasin, nous n'avons pas pu malheureusement incrémenter une option permettant de choisir le nombre d'items que l'on veut acheter pour un même article. Celui-ci est de base mis à 1. Pour acheter 2 fois le même article ; il faut faire 2 fois l'achat.

Toutes ces fonctionnalités restent donc des perspectives d'évolution.

## 0.2 Choix de mise en forme

L'arrangement du projet était assez simple. Nous avons divisé chaque page ou chaque fonctionnalité en un fichier .php, .css et .js. Cela permet une meilleure manipulation du code et une flexibilité de mise en page avec les .css surtout. Nous avons essayé de découper au mieux le CSS pour pouvoir modifier chaque fonctionnalité assez simplement sans avoir à changer tout le code.

Le nombre important de fichiers nous a incité à trier ces derniers. En effet, pour remédier à ce problème, nous avons fait le choix de créer un dossier contenant tous les fichiers .php dans lequel nous avons placé un sous-dossier avec : les fichiers .css ainsi qu'un dossier qui contient les images utilisées et les .js. Les noms donnés aux fichiers permettent d'identifier rapidement à quoi ceux-ci se rapportent.

### La partie esthétique

Pour le choix de l'aspect graphique du projet, nous avons opté pour un aspect assez "nature" avec des bords arrondis et quelque chose de pas trop "droit" pour coller au mieux aux exigences esthétiques de Mme Lacouture. La taille imposante du travail qui nous a été confiée nous a obligé à nous former en plus des cours que nous avons eu, mais le grand nombre d'informations disponibles sur internet nous a permis d'apprendre vite à retrouver l'information cherchée. La plus grande difficulté était l'organisation du code et la mise en commun des codes de chacun, codes qui ne devaient pas entrer en conflit.

### 0.2.1 Les difficultés rencontrées

Tout au long du projet, nous avons rencontré plusieurs difficultés. Nous allons diviser cette partie en citant les difficultés rencontrées dans les différents langages utilisés.

#### Les fonctionnalités

Il est parfois difficile de lier une idée esthétique à un code concret. Le plus difficile a en effet été de savoir comment donner au site une interface particulière demandée par la cliente. De plus, certaines fonctionnalités nous ont paru trop compliquées à réaliser, comme par exemple la carte interactive qui propose des activités proches du lieu d'adresse du client. Nous avons d'abord pensé à Google maps mais ceci était malheureusement très complexe. Nous avons donc décidé d'utiliser son homologue open source : OPENSTREETMAP. Pour mettre un historique des activités réalisées par le client, il aurait fallu créer une base de données qui récupère toutes les actions du client ; ce qui est assez compliqué à organiser. De plus, il faudrait faire le lien entre les différentes informations personnelles du client et les activités proposées afin de conseiller au mieux l'utilisateur sur les activités auxquelles il serait potentiellement amené à participer. Il est assez délicat, par ailleurs, de créer un espace commentaires "vivant" dans lequel tout le monde pourrait voir les commentaires des autres et où il faudrait pouvoir modérer et répondre à ces commentaires.

## HTML,CSS,JS

La difficulté de ces langages est surtout l'organisation. Il est parfois difficile de savoir de quelle manière codée pour que la page ressemble parfaitement à ce que l'on souhaite afficher, mais aussi d'éviter de se perdre dans le code, CSS en particulier, et d'y ajouter une lisibilité avec un bon découpage entre chaque partie pour qu'il soit facile d'apporter des modifications. De plus il faut optimiser le code au mieux afin d'éviter que les fichiers soient trop lourds et que le site devienne lent. L'appellation des classes de balises est aussi très important car c'est un choix crucial pour la compréhension.

## PHP,SQL

La difficulté est de bien comprendre ces langages car ils ne sont pas vraiment faits pour du traitement de texte ou bien de la mise en page. En effet, php par exemple ressemble plus à un langage "classique" de programmation, alors que SQL, de son côté, est plus un langage spécialisé dans la manipulation de bases de données. Les commande Sql sont implémentées dans le code php dans le but de manipuler des données. Il faut lors de la création de nos bases de données, réfléchir à l'organisation de celles-ci ; ce qui est l'une des plus grandes difficultés. Pour s'aider, on se pose quelques questions indispensables : Quels sont les champs importants ? Quelles seront les clefs primaires ? Quels champs allons nous mettre ? Autant de questions qui compliquent la réalisation d'une base de données.

## 0.3 Répartition du travail

Dans cette partie, nous allons aborder le sujet de la répartition du travail au sein du groupe. Au début, il n'y avait pas vraiment de répartition. Chacun codait la page ou bien la fonctionnalité qu'il sentait être importante. Sauf que par la suite, nous avons réalisé l'importance d'une organisation afin d'éviter les répétitions et les soucis de concordances entre certaines pages ou logique de code.

### 0.3.1 Répartition simplifiée

Nous avons tout d'abord essayé de répartir par tâches, c'est à dire par réalisation de pages. Voici une répartition simplifiée des tâches au sein du groupe :

	Romain	Ryan	Lilian	Nathan
<i>coté client</i>	15%	35%	15%	35%
<i>coté serveur</i>	45%	5%	45%	5%
<i>Projet KUNU</i>	30%	20%	30%	20%

La répartition sous forme de pourcentage est approximative. Elle est juste là pour indiquer de façon très brève la contribution au projet de chacun. Nous nous sommes beaucoup entraides pour résoudre les problèmes que chacun rencontrait. Chacun proposait des améliorations afin que nous soyons tous d'accord. Nous avons donc tous touchés ou ajoutés une critique à chaque partie du site, nous mettant d'accord parfois même grâce au vote à main levée.

## 0.4 Bilan

En conclusion, nous pouvons dire que ce projet nous a beaucoup apporté. Nous avons pris un plaisir à réaliser ce projet. Il nous a appris à faire des recherches plus précises pour pouvoir résoudre les difficultés rencontrés. La plus grande complexité fut le nombre important de langages différents à utiliser et le grand nombre de possibilités pour arriver à un même résultat dans certains cas.

Le projet nous a pris beaucoup de temps que nous avons très souvent partagés ensemble. Il nous a permis de créer une cohésion de groupe, de l'entraide et du dépassement de nos capacités. Nous avons essayé au mieux de soigner les détails, pour répondre au plus proche aux attentes de notre cliente. Le seul regret pourrait être le manque de temps pour pouvoir apprendre encore plus et essayer de créer de nouvelles fonctionnalités sur le site comme par exemple des propositions d'activités personnalisées .

Nous avons beaucoup aimé partager ce projet ensemble et sommes contents du résultat obtenu.