



Université de Toulouse

MASTER 2 Géomatique

« **Sciences Géomatiques en environneMent et Aménagement** »
(**SIGMA**)

Rapport de projet

Prototypage d'un site internet pour visualiser des récits d'esclaves américains

geotests.net/terrains/map

Florent LAIDIN

Commanditaire du projet : **Marie-Pierre Baduel**

Enseignant-référent : **Laurent Jégou**

Durée du projet : 15 janvier – 24 Février 2023

Table des matières

Introduction	1
1. Lancement du projet	2
1.1. Prise en main des travaux précédents	2
1.2. Début du développement	2
2. Phase de développement de l'application web	3
2.1. Utilisation du framework « Codeigniter »	3
2.2. Import de données dans MySQL	4
2.3 Méthode mise en place pour la programmation web	5
2.3.1. Création des pages	6
2.3.2. Ajout de contenu	6
2.3.3. Développement de fonctionnalités avancées	6
3. Conclusion	8
3.1. Retour d'expérience	8
3.1.1. Réalisation technique	8
3.1.2. Retour sur la gestion de projet	9
3.2. Perspectives d'évolution	10
Ressources utilisées	11
Annexes	12

Introduction

Le sujet traité dans le cadre de cet atelier de géomatique correspond au prototypage d'un site internet permettant de visualiser les récits d'esclaves américains.

Ce projet est né d'une demande concrète de Madame Marie-Pierre Baduel, professeur agrégée d'Anglais et doctorante au laboratoire CAS (Culture Anglo-Saxonne). Le sujet de sa thèse étant la « réinterprétation des récits d'esclaves au travers au prisme des images du sang et de la couleur », la visualisation cartographique des différents récits et du trajet des esclaves au cours de leur vie prend ici tout son sens.

De ce fait, l'objectif de cet atelier était de proposer un prototype de site et une étude de conception de celui-ci afin de détailler le travail effectué et ainsi garantir une meilleure continuité pour la suite du développement du site.

Pour ce travail de programmation web, j'ai pu m'appuyer sur deux projets effectués en amont. Le premier correspond à une maquette du site effectuée par des étudiants de licence d'informatique à l'Université Toulouse III. Cette maquette très précise a été réalisée selon les souhaits de Mme Baduel avec une description très technique de toutes les fonctionnalités que devait contenir le site.

Un autre projet a été effectué par les étudiants de M1 du Master SIGMA. Celui-ci avait pour objectif de construire un SIG pour les 53 récits d'esclaves avec les différentes couches propres à chaque récit. Cela signifie des frontières américaines propres à chaque période, des lieux d'esclavage, de vie, décès, naissance, pouvant être des villes (ponctuels) ou des régions, pays (polygones). Ce second projet a donc permis la création des données spatiales nécessaires au développement du site.

Le prototypage du site et son développement viennent donc en troisième partie et clôturent ce projet. Ce rapport décrit de façon synthétique le cheminement du projet et la trame de mon raisonnement qui a guidé ces 5 semaines. Il est accompagné d'un guide d'utilisation du site, et enfin d'un guide de développement sur ce qu'il reste à développer et les méthodes qui permettent d'y parvenir. Ces trois documents donnent ainsi une vue d'ensemble sur le travail effectué, les éléments réussis, les limites du prototype et les éléments à approfondir.

1. Lancement du projet

Le lancement du projet peut être décrit en deux étapes avec dans un premier temps la lecture et prise en main des différents travaux décrits en introduction. Ensuite des premiers tests ont pu être réalisés.

1.1. Prise en main des travaux précédents

Ce projet étant directement lié à deux autres travaux précédents, une phase conséquente de prise en main de ceux-ci fût nécessaire, afin de s'approprier le sujet. En effet, les travaux sur la conception du site et le guide technique ont d'abord été parcourus. Une maquette sur le logiciel « figma » a été réalisée donnant une idée du style du site et de la disposition des différentes pages.

Dans un second temps, j'ai pu charger et visualiser le SIG final conçu par les M1 avec l'ensemble des données spatiales à disposition. Là aussi, les différents styles de couches ont déjà été réfléchies permettant un gain de temps sur la visualisation du site internet.

1.2. Début du développement

Une fois les différents travaux et données parcourus, des premiers tests ont été réalisés d'abord via un serveur local : MAMP. En effet, la réalisation d'un site nécessite l'hébergement dans un serveur, de même que les données utilisées. Dans ce sens, une première solution locale a pu être effectuée avec MAMP. Ce dernier, une fois installé, permet d'avoir un serveur Apache et l'accès à MySQL.

Ce système de gestion de bases de données relationnelles est à privilégier lors du développement d'une application web. Les bases de données sont directement présentes dans le serveur, les interactions avec le code PHP sont de ce fait bien plus faciles et les chargements de données plus rapides. Si le serveur MAMP a rapidement été remplacé par l'hébergement sur le serveur Sigma, la mise en place d'une solution locale peut toujours s'avérer intéressante en cas de panne du serveur ou d'autres problèmes de connexion.

À partir de ces éléments, des premiers tests de connexion et d'affichage de données ont été réalisés sur le serveur. Les différentes pages et premiers styles ont commencé à être mis en place. Très rapidement, j'ai pu me rendre compte de la redondance de certaines informations dans le code et surtout de certaines pertes de temps lorsque l'on crée de nouvelles pages en réitérant certaines informations (appel de librairies, barre de navigation...etc). Ainsi, je me suis tourné vers l'utilisation de framework et notamment le framework php « Codeigniter » suite à une rapide formation en début de projet.

2. Phase de développement de l'application web

Suite aux premiers tests et au nombre de pages commençant à augmenter, l'utilisation d'un framework s'est avérée plus efficace pour la suite de l'atelier.

2.1. Utilisation du framework « Codeigniter »

Le framework php « Codeigniter » est basé sur le motif d'architecture logicielle « Modèle-Vue-Contrôleur (MVC) ». Ainsi le développement de l'application se divise en trois parties distinctes. Les modèles vont gérer les différentes tables MySQL et exécuter les requêtes. Les vues vont retourner les informations que l'on voit sur la page. Enfin, les contrôleurs vont permettre de guider vers de nouvelles routes et ainsi de nouvelles pages tout en envoyant les données des modèles vers les vues.

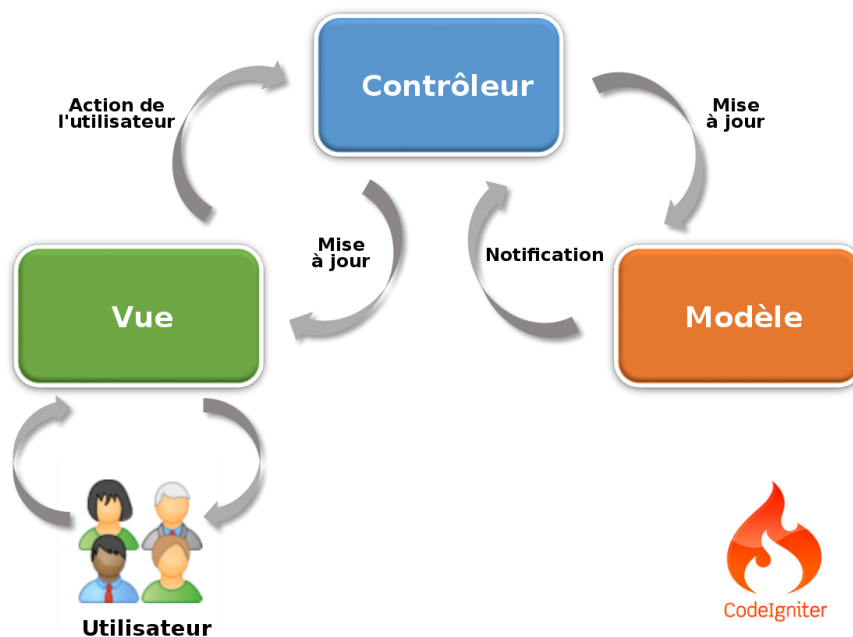


Figure 1 : Schéma du fonctionnement d'un framework sur le principe MVC

On comprend bien ici l'intérêt d'un tel outil lorsque le site devient conséquent. Le code va ainsi être divisé dans ces 3 groupes ayant chacun un rôle défini. De plus, l'utilisation d'un contrôleur va permettre d'afficher pour chaque page plusieurs vues compilées à la suite, comme montré ci-dessous.

```

28
29     public function about()
30     {
31
32         return view ('templates/header_resc')
33             . view ('resclaves/about_resc')
34             . view ('templates/footer_resc');
35     }
36
37     public function contact()
38     {
39
40         return view ('templates/header_resc')
41             . view ('resclaves/contact_resc')
42             . view ('templates/footer_resc');
43     }
44
45

```

Figure 2 : Illustration de fonctions utilisés dans le contrôleur Map.php

Comme on peut le voir ici, deux fonctions sont écrites renvoyant chacune plusieurs pages php. Par exemple, si l'on prend la fonction about(), la page renvoyé sera composé de trois fichiers php dont un header et footer qui peuvent être appelés dans d'autres pages. Ainsi si l'on veut modifier certains imports de librairies ou une barre de navigation, il suffit de le faire dans un seul fichier et non dans toutes les pages.

Pour terminer, nous pouvons dire ici que le développement du site s'est donc effectué avec l'utilisation combinée de CodeIgniter et de MySQL pour une meilleure optimisation du code.

2.2. Import de données dans MySQL

Bien que MySQL dispose de nombreux avantages dans la mise en place d'application web et de les relier avec les données, la prise en compte des données spatiales est plus délicate. En effet, les données de type 'geometry' ne sont pas supportées par MySQL, à l'inverse de PostgreSQL qui dispose de son extension spatiale PostGIS.

Pour ce projet, l'import des données spatiales et leur affichage sur le web a nécessité plusieurs traitements en amont. Les données spatiales ont pu être importées avec des géométries au format WKT qui est un format texte. L'export des données a donc été effectué sur QGIS en CSV avec un champ géométrie en WKT. Une fois la table importée dans MySQL, un champ a pu être créé en transformant le WKT en coordonnées GeoJSON, grâce à la fonction ST_AsGeoJson() comme visible ci-dessous.

```

UPDATE points
SET geoj = SELECT ST_AsGeoJson(GeomFromText(wkt,4326)) AS geoj
FROM points ;

```

Une fois ce champ créé, il était alors possible de générer un GeoJSON via la librairie Leaflet.js et du code PHP.

Pour terminer sur ce point de l'import des données dans MySQL, il est important de rappeler comment étaient structurées celles-ci au sein du SIG conçu par les M1. Les données spatiales étaient séparées chacune dans un second groupe correspondant au récit. L'agrégation de toutes les couches de type point et de toutes celles de type multipolygone a donc nécessité quelques traitements.

Les couches étant nommées de la même manière, une boucle python (annexe 1) parcourant l'ensemble des dossiers a pu être effectuée pour les copier dans un même répertoire et les afficher dans QGIS. Un identifiant récit a ensuite pu être généré pour chacune des couches, celui-ci correspondant aux deux premiers chiffres du récit (01, 02...etc).

2.3 Méthode mise en place pour la programmation web

Suite à la présentation du framework et du système de gestion de bases données utilisés, il peut être intéressant de détailler la méthode mise en place pour développer le site et ses différentes pages.

Ici je me suis appuyé sur les travaux des étudiants en licence d'informatique avec la maquette et l'arborescence du site (figure 3). La méthode peut être détaillée en deux phases avec dans un premier temps la création du « squelette » de l'application web et dans un second l'ajout de contenu.

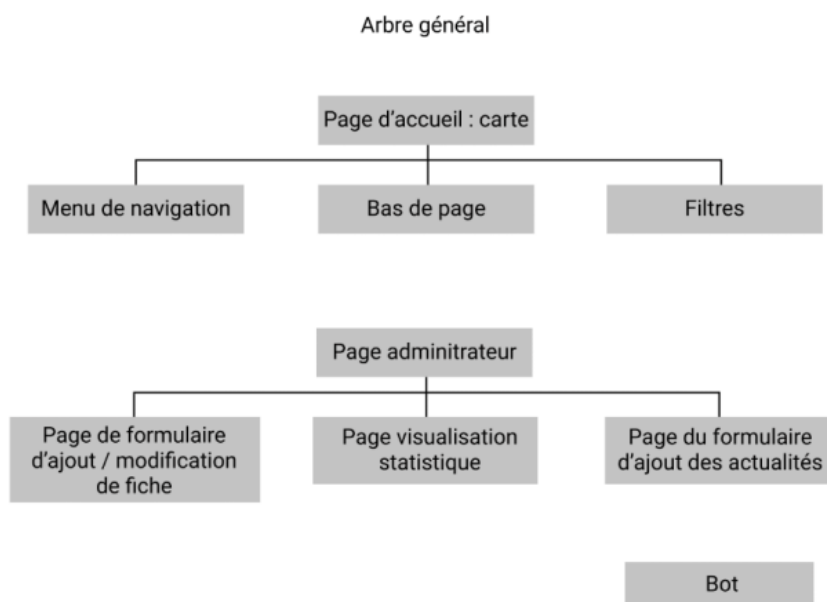


Figure 3 : Arborescence générale attendue du site ; Source : cahier des charges du site esclaves (étudiants L3 informatique)

2.3.1. Création des pages

En premier lieu, la création d'un site web nécessite de créer différentes pages reliées entre elles et pouvant accepter de nouvelles données. L'utilisation de CodeIgniter est ici essentielle. Différentes routes ont pu être créées pour rediriger l'utilisateur aux pages souhaitées. A partir de ces nouvelles routes, des fonctions basiques ont été créées dans les Contrôleurs pour afficher les vues.

Une fois ces éléments mis en place, on dispose d'un site, certes sans contenu, mais qui contient déjà différents templates, prêts à être développés.

2.3.2. Ajout de contenu

Pour ce qui est de l'ajout de contenu, je me suis d'abord concentré sur les parties « header » et « footer » du site. En effet, ce sont les parties structurantes de chaque page. Comme indiqué dans le guide d'utilisation, ces deux éléments vont contenir des liens vers d'autres pages permettant à l'utilisateur de naviguer plus facilement dans le site. Ces éléments de navigation ont pu être effectué grâce au framework Bootstrap CSS.

Ensuite, la page d'accueil (/map) a pu être complétée avec l'ajout de l'élément carte et du panneau latéral. Des premières données provenant de la base MySQL ont pu être ajoutées, il s'agit des points de publication des récits. Ces derniers apparaissent dès l'appel de la page d'accueil sous forme de clusters, étant donné que de nombreux récits ont été publiés au même endroit.

Viennent ensuite les informations sur la liste des récits au sein d'un tableau (/recits). Cela correspond au second lien effectué avec la base de données MySQL. Les données provenant de la table des récits sont appelées au sein d'un tableau pour chacun des 53 récits avec plusieurs informations comme le nom de l'esclave, la date de publication, ou bien le titre du récit, par exemple.

2.3.3. Développement de fonctionnalités avancées

Une fois ces premiers éléments effectués, l'ajout d'un premier filtre au sein du panneau latéral a été mis en place. Il s'agit d'un filtre permettant la sélection d'un récit en particulier. Une fois validé, les différentes couches en lien avec le récit apparaissent avec notamment les frontières étatsuniennes à la date de publication mais également les différents lieux de publication, décès, naissance. À noter qu'on ajoute également deux couches pérennes que sont les royaumes africains et les aires autochtones amérindiennes.

Concrètement, cela s'est effectué en plusieurs étapes avec dans le menu déroulant l'ajout d'une valeur propre à chaque récit (id_recit). Cette même valeur est récupérée dans le contrôleur qui lui-même va appeler une fonction d'un modèle pour chercher l'ensemble des données qui contient cette valeur. Ce principe est illustré en figure 4.

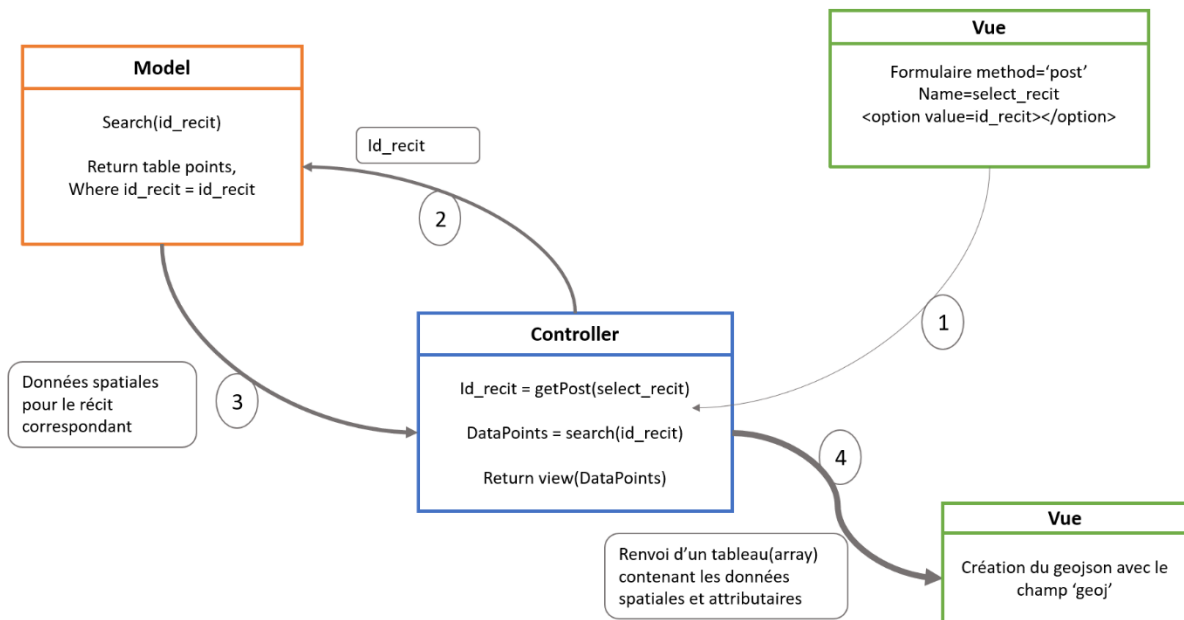


Figure 4 : Fonctionnement schématisé d'un filtre lors du choix d'un récit

Un second filtre fonctionnant sur le même principe a pu être ajouté. Celui-ci a permis d'afficher les récits par types de lieu. C'est-à-dire les lieux de naissance, décès, d'esclavage ou bien les lieux de vie après l'esclavage.

Pour ce qui est de la liste des récits (/recits), on a tout d'abord un tableau html écrit avec une boucle php. Pour les 53 récits de la table, ceux-ci vont être contenus dans le tableau avec le nom de l'esclave, la date de publication et le titre. L'utilisateur a la possibilité de visualiser plusieurs informations au sein de la fiche récit en cliquant sur le nom de l'esclave.

À noter ici que cette fiche récit est disponible dès la page d'accueil si l'utilisateur clique sur les points. Un lien vers chacun des récits est effectué dans les pop-ups. Ces liens vers la fiche ont été possible grâce à la création d'une nouvelle route (/recits/(:segment)) selon le récit sélectionné. Ainsi, chacun des récits dispose d'une url unique qui le redirige vers sa fiche.

3. Conclusion

Pour conclure ce rapport de projet, un retour d'expérience sur les réussites et principales difficultés peut être effectué, suite à quoi viendra une discussion sur la suite de cet atelier.

3.1. Retour d'expérience

3.1.1. Réalisation technique

Tout d'abord, ce projet a permis la réalisation d'un site fonctionnel et accessible sur le web. En effet, de nombreux éléments initialement souhaités par la commanditaire ont pu être réalisés. J'entends par là que le site web contient à présent deux pages principales correspondant chacune à un détail des récits, qu'il soit cartographique ou selon d'autres critères attributaires. Des liens ont pu être effectués entre les pages, comme détaillé dans le guide d'utilisation, rendant ce site dynamique.

D'un point de vue cette fois-ci plus technique, la mise en place d'une application web fonctionnant grâce à un framework me paraît être une réussite, notamment dans l'idée ou celui-ci va être finalisé par d'autres personnes. Les différents fichiers sont bien distincts et présents dans différents répertoires. Il s'agit alors de regarder les différentes routes puis le contrôleur et on comprend très rapidement quels sont les fichiers à modifier. Il s'agit donc d'une solution plus robuste et réutilisable.

Il est important de préciser que ce prototype n'a pas été fait sans difficulté. En effet, je me suis heurté à plusieurs problèmes lors du développement du site, notamment sur la partie cartographique et les injections de données au travers de CodeIgniter.

Comme énoncé plus haut, les données ont été importées avec des champs de géométrie au format WKT (text). Si cela ne pose pas de difficultés pour des géométries légères de type « point », c'est un peu plus délicat pour les multipolygones. Pour pouvoir appeler les couches correspondantes à chaque récit lors de leur sélection sur la page d'accueil, j'ai créé une seule table pour les points et une autre pour les polygones. Or, cela rassemble en une même table un volume de données considérable. Plusieurs opérations ont donc été nécessaires comme la simplification des géométries sur plusieurs kms et l'import par fichier dump SQL dans la base de données. De plus, comme dit précédemment, le rassemblement des données par type de géométrie a pu prendre un certain temps.

La seconde difficulté majeure englobe la gestion des styles de couche de manière « propre ». L'affichage des couches ayant pris un certain temps, la gestion de tous les styles et de leur affichage (arrière-plan, couleurs...etc) est arrivée sur le tard. Ainsi, les styles actuellement présents au sein du site ont pour la plupart été réalisés dans des fonctions javascripts sous forme de conditions multiples. Cela n'est pas la solution optimale si l'on veut changer certains paramètres. À cela je souhaitais rajouter que les styles des pays représentant plusieurs lieux (par exemple, naissance, esclavage et lieu de vie) est resté en suspens, la charte de graphique proposé par le groupe de M1 est difficilement réalisable.

3.1.2. Retour sur la gestion de projet

Je peux terminer ce retour d'expérience sur les aspects de gestion de projet. En effet, la réalisation d'un tel site web en quelques semaines était un objectif ambitieux mais atteignable. Les premières semaines ont été un peu plus laborieuses en raison notamment d'un manque de méthode et d'organisation. Les premiers tests réalisés étaient un peu désorganisés. C'est-à-dire que j'essayais de faire de nombreuses choses en même temps, comme l'import de données spatiales mais également la mise en place de filtres.

L'utilisation de Codelgniter a permis beaucoup de choses mais il est important de préciser que la prise en main complète de cet outil peut prendre un certain temps. En cela, la formation sur Codelgniter réalisée en troisième semaine de projet a été un vrai plus pour mieux comprendre l'utilité de ce framework. J'ai pu alors réellement commencer à mettre en place les éléments au fur et à mesure et à faire des tests.

Le développement de ce site est en effet passé par un aller-retour permanent entre le développement d'une nouvelle fonctionnalité, d'un nouveau style et l'actualisation de la page. C'est un travail qui est très chronophage, le seul affichage de points sous la forme de clusters avec des couleurs personnalisées peut prendre plusieurs heures.

C'est pourquoi je me suis d'abord concentré sur l'aspect fonctionnel des grands objectifs du projet, à savoir les deux pages principales, la possibilité de visualiser la fiche de chaque récit et également d'avoir un filtre par récit sur la carte, suite à quoi sont venues les autres fonctionnalités en fin de projet.

Les réunions hebdomadaires avec la commanditaire (Mme Baduel) et mon encadrant (M. Jégou) ont toujours été riches de contenu avec à la fois des discussions sur les aspects techniques de la programmation web et sur le rendu du site. Ces réunions ont bien souvent permis d'améliorer certaines fonctionnalités ou styles d'affichage rendant bien agréable l'expérience utilisateur. En effet, le fait de concevoir et programmer le site rend parfois moins alerte à la logique de certaines fonctionnalités. En cela, ces réunions permettaient directement de savoir si le fait de cliquer à certains endroits paraissaient évidents ou si ça ne l'était pas du tout.

Ensuite, j'émet un léger regret quant au fait de ne pas avoir fait la réunion avec la commanditaire en semaine 4, étant donné que Mme Baduel aurait pu dès cette fois-ci indiquer de nombreux éléments à reprendre. Je pensais que l'avancement minime durant cette semaine n'aurait pas soulevé autant de remarques et conseils.

Enfin, j'ai pu compter sur la réactivité de M. Jégou lorsque j'avais des questions ou difficultés. Il est important de préciser ici que le serveur sigma a fermé au milieu de l'atelier suite à quoi le prototype a pu être hébergé sur le serveur de M. Jégou. Cela a permis de sauver plusieurs jours de projet.

3.2. Perspectives d'évolution

Pour terminer, ce site est encore un prototype. Cela signifie que de nombreux éléments sont à retravailler et créer. Ceux-ci seront développés plus en détail dans le guide de développement mais nous pouvons déjà citer quelques grandes thématiques.

Comme énoncé précédemment, une gestion des styles au sein de fichiers CSS est préférable de manière à pouvoir les changer plus facilement et proprement. L'ensemble des styles des GeoJSON est en effet contenu dans des fonctions rassemblant tout un ensemble de conditions.

Toujours sur l'aspect cartographique, d'autres filtres sont à prévoir et notamment le filtre sur le type de récit (écrit ou dicté) que je n'ai malheureusement pas eu le temps de développer sur la fin du projet.

Au niveau de la visualisation des récits et de leurs caractéristiques, des filtres peuvent être mis en place pour rechercher certains récits en particulier comme par exemple la date de publication, le type de récit. La fiche de chaque récit est également à retravailler puisqu'elle est actuellement très sommaire.

Enfin, l'objectif final de cette commande est aussi d'avoir une gestion des authentifications pour que les personnes enregistrées aient la possibilité d'ajouter des récits, de manière contrôlée et sécurisée.

Ressources utilisées

- Cahier des charges du site « Slave Narrative », Emilien NIMAL, Dany NEANG, Fabien FOCK YING CHU, Khaled AL MUGHRABI, Valentin HOUDOU, Mohammed JARALLAH
- Maquette figma du site Slave Narratives ; URL : <https://www.figma.com/proto/vJEdJa5tKpJJkItZOSveQv/Projet-final?node-id=144%3A19964&scaling=contain&page-id=0%3A1>
- Projet SIG contenant l'ensemble des données spatiales et styles nécessaires pour le site internet, Master 1 SIGMA.
- Charte graphique des couches utilisées dans le projet SIG, Master 1 SIGMA.
- Tutoriel du framework PHP CodeIgniter : https://codeigniter.com/user_guide/index.html
- Exemple d'application web par CodeIgniter : site de visualisation des terrains du LISST.

Annexes

Annexe I

Boucle python récupérant l'ensemble des couches de type points au sein de différents dossiers

```
import os
import shutil

wd = r"D:/M2/atelier/PROJET_QGIS_M1/COUCHES/COUCHES_ETAPES_RECITS/"
os.chdir = wd
list_of_files = os.listdir(wd)

dest_dir = r"D:/M2/atelier/points/"

for direct in list_of_files:
    path =
    r"D:/M2/atelier/PROJET_QGIS_M1/COUCHES/COUCHES_ETAPES_RECITS/"+direct
    print(path)
    os.chdir = path
    list = os.listdir(path)
    for f in list:
        if f[4:6] == "pt": # If file contains « pt »
            print(f)
            filePath = shutil.copy(path+'/' +f
                                   ,r"D:/M2/atelier/points/" )
```