



## LO07 –Technologie du web

Participants :

- Malo DE RIVALS ISI2
- Antoine GRUCHET ISI1

### Résumé :

Dans le cadre de l'UV LO07 nous avons dû effectuer un projet de grande envergure  
*Dans ce rapport, nous vous expliquerons en détails la réalisation de ce projet de sa création a son aboutissement.*

Responsable de l'UE : Marc Lemerancier

Semestre : Printemps 2017

## Sommaire

<b>SOMMAIRE .....</b>	<b>2</b>
<b>INTRODUCTION .....</b>	<b>3</b>
<b>REMERCIEMENTS .....</b>	<b>4</b>
<b>ANALYSE .....</b>	<b>5</b>
<b>BDD .....</b>	<b>5</b>
1) LES TABLES.....	5
2) LES CONTRAINTES DE CLE ETRANGERE .....	5
.....	6
<b>GIT ET GITHUB.....</b>	<b>7</b>
<b>FONCTIONNALITES .....</b>	<b>7</b>
A. CLASSES ET CLASSES MANAGER.....	7
B. AUTHENTIFICATION INSCRIPTION .....	8
C. CREER CURSUS (PERSISTANCE).....	9
D. VISUALISER CURSUS.....	10
E. IMPORTER/EXPORTER CURSUS .....	10
F. ANALYSER CURSUS .....	11
<b>BOOTSTRAP .....</b>	<b>11</b>
<b>JAVASCRIPT.....</b>	<b>12</b>
<b>DIFFICULTES RENCONTREES .....</b>	<b>12</b>
<b>LES APPORTS PERSONNELS .....</b>	<b>12</b>
<b>CONCLUSION.....</b>	<b>14</b>
<b>ANNEXE .....</b>	<b>15</b>
BASE DE DONNEES (AUSSI DISPONIBLE DANS LES FICHIERS DU PROJETS) : .....	15

## Introduction

Nous avons réalisé un projet informatique tout au long du semestre, celui-ci a pour but de nous familiariser avec le langage informatique lié au web, nous nous sommes donc familiarisés avec les langages HTML, CSS, PHP et JAVASCRIPT.

Ceci dans le cadre de notre cursus à l'UTT. Les technologies du web peuvent compliqué au premier abord, en grande partie car elles sont complexes. Cependant avec un travail régulier, certaines notions sont assimilables sur un semestre.

Nous avons fait le projet à l'aide du super éditeur de texte (IDE) Netbeans. Celui-ci nous a été d'une grande aide car il s'agit d'un logiciel très intuitif.

Pour pouvoir lire un langage serveur (le langage PHP), nous avons dû utiliser un serveur apache, l'un utilisait MAMP car il est sur Mac, l'autre utilisait Wamp car il est sur Windows. Tout au long du projet nous avons donc dû nous adapter.

Venant tous les deux de tronc commun nous découvrons les technologies du web et il s'agissait de l'un de nos premiers grands projets informatiques. Nous avons cependant certaines connaissances dans le domaine de l'informatique, en effet l'un a fait l'UE LO02, l'autre à fait NF05.

Nous expliquerons dans ce rapport ce que nous avons appris pour mettre au point un tel projet. C'est pourquoi, dans un premier temps nous verrons notre base de données, ensuite nous expliquerons brièvement les éléments de nos fonctionnalités, enfin nous expliquerons nos ajouts de code Javascript et Bootstrap.

## Remerciements

Nous remercions notre professeur de cours et de TD, mais aussi responsable de l'UE monsieur Lemerrier de nous avoir enseigné les fondamentaux des langages du web (php, html, js, sql) et son aide au cours des TDs et du projet.

Nous remercions aussi madame Deschamps pour son aide au cours des TDs et pour le projet.

Et enfin nous remercions monsieur Mordefroy pour son attention et son évaluation attentive.

## Analyse

L'objectif de ce projet est de concevoir, de développer et de tester un site Web dynamique pour la gestion des cursus des étudiants du programme ISI de l'UTT. Le sujet fut tout d'abord complexe pour nous a bien comprendre, nous avons dû prendre et reprendre le sujet de fond en comble. Avant toutes choses il fallait comprendre les liens entre tous les éléments du projet, quel est l'objectif ?

Une des choses les plus difficiles était d'avoir une vue d'ensemble pour bien démarrer, ceci commence par beaucoup de brainstorming, de discussions. Il fallait ici que chacun expose sa compréhension du sujet. Nous avons commencé doucement, ceci nous a permis d'avoir des spécifications du problème correctes.

Même si nos spécifications présentaient encore quelques éléments flous, nous nous sommes lancés dans la mise en place de notre base de données.

## BDD

### 1) Les tables

Notre base de données comprend cinq tables comme nous pouvons le voir sur le schéma ci-dessus. Chaque table dispose d'un attribut avec une clé primaire afin de garantir l'unicité de chaque élément rentré dans la base de données et de pouvoir récupérer les informations des cursus ou des règlements sans se tromper.

La clé d'un règlement ou d'une règle dépend du label indiqué dans la première ligne du règlement en question. En effet ce label est récupéré et devient le nom (unique) du règlement. Ensuite il est récupéré pour l'identifiant de chaque règle auquel on concatène le numéro de la règle en question pour assurer l'unicité de cette règle.

La clé d'un étudiant est son numéro étudiant (récupéré dans le cursus ou au moment de la création de l'étudiant sur le site). La clé d'un cursus est son label qui le nom du fichier csv importé moins l'extension. Si le cursus est créé directement sur le site alors on demande à l'étudiant de choisir le label. Le label d'une UE est le sigle de cette UE auquel on rajoute le numéro étudiant par concaténation.

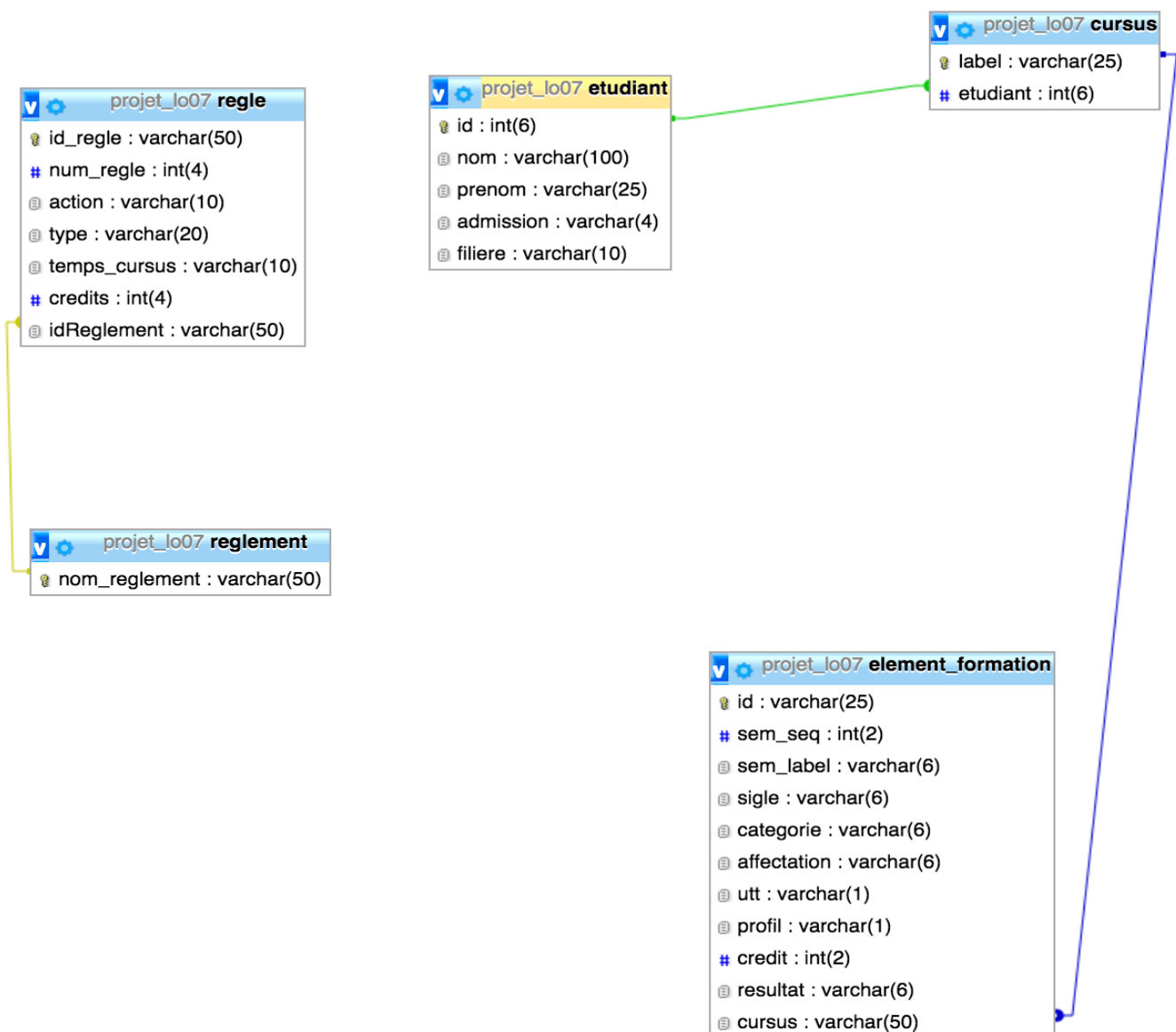
Nous avons séparé ces tables en deux groupes différents, d'un côté les règlements et règles (qui servent à vérifier si un étudiant peut être diplômé), et de l'autre les cursus, les étudiants et les éléments de formation. Pour ce faire, nous avons placé des contraintes de clés étrangères sur les attributs des tables pour les lier ensemble.

### 2) Les contraintes de clé étrangère

Ainsi le nom d'un règlement (qui est aussi l'unique attribut de cette table) est lié à l'attribut indexé 'idReglement' d'une règle. La mise en place de la table règlement peut paraître inutile au premier abord étant donné qu'elle ne contient qu'un attribut mais la clé étrangère permet de manipuler toutes les règles liées à ce règlement d'une seule commande sur son nom. De plus il est plus facile de contrôler son unicité ainsi.

Le deuxième groupe permet de lier un élément de formation à un cursus et de lier ce cursus à un étudiant. En effet ainsi, si on supprime un étudiant, le cursus qui lui est lié est automatiquement supprimé grâce à la contrainte de clé étrangère et les éléments de formation liés à ce cursus sont supprimés à leur tour.

## Concepteur de notre BDD



## Git et github

Nous sommes étudiant ingénieur en informatique, nous découvrons le monde de la programmation. Celui-ci peut s'avérer très intéressant, il nécessite une rigueur remarquable et demande d'être ordonné pour mieux réfléchir. Etant donné que nous commençons à bien comprendre les technologies du web, nous nous sommes initié a git et github avant de rentrer en profondeur dans le projet. L'intérêt de s'approprier ce logiciel de gestion de version décentralisé est important car il est indispensable pour tous les développeurs, mais aussi ce logiciel permet de bien travailler sans interférer sur le travail de l'autre.

Nous avons donc chacun une branche. Après un début fastidieux sur git et github souvent des problèmes à résoudre que nous développerons dans la partie sur les difficultés rencontrées (le logiciel appelle ça des « conflits »), nous nous sommes bien appropriés ce logiciel qui nous a été bénéfique pour optimiser le rendement de travail.

Pour bien utiliser git il a été important d'utiliser un terminal de commande et de connaître une base de commande. Il fallait connaître les commandes permettant de naviguer entre fichier (cd, ls, pwd...), les commandes permettant de créer, un dossier, un fichier (mkdir, touch...) même si nous pouvions le faire directement depuis l'IDE. Pour bien comprendre ce logiciel, nous avons suivis divers tutoriel écrit et vidéo sur internet nous expliquant les multiples commandes de git. Le principe de git est le suivant : pouvoir dans un premier temps versionner son code au cas où une modification perturbe le bon fonctionnement du site (revenir en arrière dans les versions). Puis dans un second temps il permet de pouvoir échanger notre code à l'aide des commandes git push et git pull qui permettent respectivement de déposer le déposer et de récupérer des modifications qui ont été versionner a l'aide de la commande git commit.

## Fonctionnalités

### A. Classes et Classes Manager

Après avoir créé notre base de données (BDD). Nous avons décidé d'effectuer notre projet en orienté objet, nous avons donc créé nos classes à l'image de la BDD. Nous avons ajouté des classes manager pour chaque classe présente. Ces classes Manager permettent de faire le lien avec la BDD. Ceci nous a été d'une aide colossale tout au long du projet. La programmation orientée objet est un plus pour le développeur, en effet celle-ci permet d'aller plus vite, car elle facilite la compréhension du problème traité et permet de bien faire la part des choses pour résoudre le problème.

Nous avons donc 10 classes, 5 classes permettant de traiter avec les valeurs en interne et 5 autres classes Manager. Les 5 premières correspondent aux classes étudiant, élément de formation, cursus, règlement et règle. Ces classes, l'image de leur table de même nom dans la base de données nous permettent de traiter dynamiquement avec les valeurs au sein d'une page web.

Les 5 autres classes nous permettaient d'aller chercher les valeurs dans la base de données en fonction d'un id précis. Elles nous permettaient de remplir notre base de données. D'autres fonctions très utiles sont présentes dans ces deux classes.

	Classes (etudiant, cursus ...)	Classes Manager (etudiantManager...)
<b>Fonctions</b>	Getter et Setter Hydrate (array \$donnees) __construct(\$donnees) __toString()	__construct(\$db) setDb() add(class \$class) delete (classe \$classe) get(\$id) getList()

Ce tableau résume les principales classes présentes dans le code de notre site. Il s'agit du cœur de notre code qui permet de faire transiter les valeurs des superglobales \$\_POST et \$\_SESSION.

À l'aide de la classe etudiant et de la classe etudiantManager nous avons pu mettre au point un système d'authentification et d'inscription.

## B. Authentification Inscription

L'utilisateur a les possibilités de s'identifier en rentrant son numéro étudiant, de s'inscrire s'il n'est pas déjà rentré dans la base de données et d'importer un cursus, lors de l'import s'inscrit car l'étudiant est créé.

C'est-à-dire que l'étudiant arrive sur la page d'accueil, il a alors le choix : soit il s'authentifie, soit il s'inscrit soit il importe un cursus. S'il s'authentifie, le site récupérera alors toutes ses données à l'aide des classes Manager.

S'il s'inscrit, ses données sont dans un premier temps envoyées dans la BDD, puis manipulées à l'aide de variables de session. Pour faire en sorte que l'utilisateur s'authentifie ou s'inscrive nous avons mis en place des formulaires html, dont certains éléments sont dynamiques.

Ensuite l'utilisateur est redirigé vers la page d'accueil du site (index.php)

Explication du site

[Inscrivez vous](#)

[Importer un cursus](#)

[Authentifiez-vous](#)



### C. Créer cursus (persistance)

Vient alors la réalisation de la 1<sup>ère</sup> fonctionnalité, ce fut pour nous une étape importante car c'est à ce moment que nous avons créé les instances de PDO pour faire appel à notre base de données. Cette tâche s'est avérée dans un premier temps périlleuse, cependant, les notions viennent à force de pratique. On demande dans un premier temps à l'utilisateur de créer et de nommer un cursus. Le nom du cursus est alors stocké dans la BDD, ensuite l'utilisateur doit rentrer ses éléments de formation via des formulaires html :

Ajouter l'élément de formation (UV et ses détails)

Semestre n° :

Label(ISO1..):

Sigle:

Catégorie(CS, TM):

Affectation :

☐ TC

☐ TCBR

☐ BR

Suivis à l'UTT :

☐ oui

☐ non

L'UE est-elle dans votre profil :

☐ oui

☐ non

Résultat:

Submit

Reset

Une fois l'élément de formation rentré dans la BDD, l'utilisateur a le choix soit de retourner à l'accueil pour effectuer d'autres opérations sur divers cursus ou soit de rentrer un autre élément de formation.

Sur l'index ainsi que sur de multiples pages sont disposés des boutons permettant de naviguer de page en page et de profiter pleinement des fonctionnalités du site web.

## D. Visualiser cursus

Cette fonctionnalité permet d'afficher les informations d'un cursus. Elle donne des informations sur les UE qui lui sont liées et leur résultat. Ces informations sont affichées par catégories. Il n'est pas tenu compte du semestre auquel elles ont été suivies parce que nous avons estimé que le besoin dans ce cas est de faire un point sur son cursus (ex : le nombre de CS/TM suivies, les crédits manquants, etc.) et que c'est plus facile à faire en le visualisant ainsi.

**Beatrice PRIOR, vous avez choisi le cursus cursusPrior**

**CS (18 crédits) :**

- SY02 : A (6 crédits)
- NF20 : A (6 crédits)
- NF16 : A (6 crédits)

**CT (8 crédits) :**

- SC00 : A (4 crédits)
- UX80 : EQU (4 crédits)

**HP :**

**TM (18 crédits) :**

- IF14 : A (6 crédits)
- UX51 : EQU (6 crédits)
- UX50 : EQU (6 crédits)

**EC (4 crédits) :**

- LE08 : A (4 crédits)

**ME (12 crédits) :**

- GE04 : B (4 crédits)
- UX71 : EQU (4 crédits)
- UX70 : EQU (4 crédits)

**ST :**

- TN09 : A (30 crédits)
- TN10 : A (30 crédits)

**BULATS validé.**

**Semestre à l'étranger  
effectué.**

Analyser un cursus

Accueil

Visualiser autre Cursus

Importer Cursus

## E. Importer/Exporter cursus

Cette fonctionnalité importe les fichiers au format csv. Les fichiers cursus sont découpés en deux parties. La première permet de récupérer les informations de l'étudiant concerné et de l'ajouter dans la base de données. La seconde permet de récupérer tous les éléments de formation que cet étudiant a suivis au cours de ce cursus.

Lorsque l'étudiant est enregistré dans la base de données, un cursus est automatiquement créé grâce à contrainte de clé étrangère sur son numéro étudiant. Ensuite ce cursus est lié aux éléments de formation par la contrainte sur l'index 'cursus' dans la table 'element\_formation'.

Nous avons choisi aussi de mettre en place une fonction d'import de règlement par souci de flexibilité.

La première ligne du fichier csv permet d'obtenir le label du règlement qui se doit d'être unique dans la base de données. A partir de ce label un nouveau règlement est importé dans la base de données. Ensuite on récupère chacune des règles les unes après les autres et on les ajoute à la base de données. On ajoute comme attribut le label du règlement pour les lier au règlement en question et les manipuler ainsi plus aisément.

Cette page a été conçue spécialement pour que vous puissiez déposer des cursus ou des règlements au format CSV

Choisis le cursus a importer

Aucun fichier choisi

Importer

Choisis le règlement que tu veux importer

Aucun fichier choisi

Importer

## F. Analyser cursus

Cette fonctionnalité permet de vérifier si un cursus choisi remplit les conditions d'un règlement sélectionné.

On commence par récupérer toutes les règles du règlement en question. Puis pour chaque règle on regarde tous les éléments du cursus. En parcourant tous ces éléments on vérifie qu'ils remplissent les conditions de la règle. Une fois que tous les éléments ont été parcourus on vérifie que le minimum de crédits requis a été atteint (dans le cas d'une requête « SUM ») ou que l'UE demandée a bien été validée (dans le cas d'une requête « EXIST »). Si ce n'est pas le cas alors la règle n'est pas validée et on ajoute dans un tableau une phrase indiquant ce qu'il manque pour valider cette règle.

Une fois que toutes les règles ont été analysées on retourne ce tableau. S'il est vide alors on affiche que le cursus est conforme. En revanche si ce tableau n'est pas vide, les conditions sont affichées pour montrer ce qu'il manque à ce cursus.

---

Ce cursus n'est pas conforme Antoine :

- Il manque 54 crédits de CS+TM en TCBR .
- Il manque 30 crédits de CS+TM en FCBR .
- Il manque 30 crédits de CS en BR .
- Il manque 30 crédits de TM en BR .
- Il manque 30 crédits de ST en TCBR .
- Il manque 30 crédits de ST en FCBR .
- Il manque 12 crédits de EC en BR .
- Il manque 4 crédits de ME en BR .
- Il manque 4 crédits de CT en BR .
- Il manque 16 crédits de ME+CT en BR .
- Il manque 60 crédits de CS/TM en BR à l'UTT.
- Il manque SE à votre cursus.
- Il manque NPML à votre cursus.
- Il manque 174 crédits au total.

Choisir un autre cursus

Accueil

Visualiser Cursus

Importer Cursus

Bootstrap

Bootstrap est un framework CSS permettant de mettre en page notre site web à l'aide de grilles pour bien positionner les paragraphes de notre page.

Il s'agit d'un ensemble de composants structurés qui sert à créer les bases et à organiser le code informatique pour faciliter le travail des programmeurs, que ce soit en termes de productivité ou de simplification de la maintenance. Il en existe beaucoup pour les applications web qui ciblent de nombreux langages : Java, Python, Ruby, PHP...

Pour pouvoir utiliser Bootstrap, il suffit de le télécharger et d'appeler le fichier CSS correspondant en haut de la page html. Bootstrap a su nous montrer son efficacité dans la bonne présentation de nos formulaires.

Pour mener à bien notre projet nous avons dû nous familiariser avec Bootstrap.

## JavaScript

Un des soucis majeur que nous avons rencontré est l'erreur de frappe. Lorsque nous rentrons un élément de formation, si celui-ci est mal rentré, il peut interférer avec toutes les autres fonctionnalités. JavaScript nous permet de fixer des conditions pour s'assurer que le champ du formulaire est bien rempli avec les bonnes spécifications.

## Difficultés rencontrées

Au cours de ce projet nous avons bien évidemment rencontré de nombreuses difficultés, à commencer avec git et github. En effet nous avons dû recommencer à plusieurs reprises notre projet pour cause d'erreurs de manipulations sur git. De plus sur la fin du projet, l'utilisation de la commande « git merge » sur la branche master avec les deux autres branches a engendré des conflits que nous avons cru résoudre correctement alors que certaines lignes de code ne correspondaient pas à la bonne version. Le problème étant que les différences étaient minimes nous avons donc mis du temps à comprendre l'origine du problème.

Etant donné que nous travaillons respectivement sur Mac et sur Windows, le logiciel pour la base de données était MAMP pour le premier et WAMP pour le second, ce qui entraînait un réglage des propriétés différent pour chacun. Mais à chaque mise à jour de la branche de l'autre, les propriétés étaient modifiées aussi et il fallait les rétablir.

Au tout début nous avons fixé les contraintes de clé étrangère dans le mauvais sens, ce qui posait plusieurs problèmes pour les imports des différents éléments.

Mais ce qui a été le plus compliqué au cours de ce projet, c'est de s'approprier le code l'un de l'autre. En effet, lorsqu'on se retrouvait à utiliser une fonction codée de notre binôme, il fallait bien contrôler tous les cas rendus possibles par cette fonction. Pour ce faire l'utilisation de « var\_dump(\$expression) » nous a permis d'analyser le code en divers endroits jusqu'à comprendre l'utilité de chaque ligne.

## Les apports personnels

Ce projet nous a beaucoup apporté personnellement. En effet chaque projet à l'UTT nous forme un peu plus à gérer notre temps au cours du semestre pour ne pas être débordé à la fin quand les examens approchent et que tous les projets doivent être menés à terme.

Mais aussi les contraintes imposées par le cahier des charges nous ont amené à nous répartir le travail pour répondre aux différents besoins du projet. Ainsi nous avons pu améliorer notre autonomie et le travail en équipe en même temps. Nous avons donc pu apprécier l'utilité de ces conseils élémentaires :

- commenter son code pour le rendre plus lisible et compréhensible pour le prochain, ou même pour soi quand on revient après quelques temps.
- Nommer ses variables de manière pertinente dans le contexte.

Ce projet nous a aussi permis d'acquérir une bonne base de connaissances en php, html, sql, etc. Rien ne vaut la pratique et quelques moments de réflexion intense sur un bout de code qui pose problème pour assimiler les particularités du php.

## Conclusion

L'UE LO07 est avant tout une UE très ludique surtout pour des étudiants qui ne connaissent pas le monde du web. Le cahier des charges nous pousse à chercher des solutions par nous-mêmes et sur internet. C'est en butant sur des problèmes tout au long du projet que nous avons dû rechercher de nouvelles ressources et approfondir les connaissances déjà apportées par le cours.

Git et github sont fondamentaux pour les développeurs, Bootstrap est le framework CSS le plus utilisé aujourd'hui. Nous avons eu la chance de pouvoir apprendre, grâce au projet, des éléments prépondérants dans le monde du web.

Nous avons été tout au long de ce projet en constante découverte. Après un projet de cette envergure nous sommes capables d'effectuer un stage de développeur web.

Pour conclure nous pouvons dire que cette UE nous a apporté tant sur le plan professionnel que sur le plan culturel, surtout parce que un projet comme celui-ci apporte en rigueur et en connaissance.

## Annexe

Base de données (aussi disponible dans les fichiers du projets) :

```
-- phpMyAdmin SQL Dump
-- version 4.6.4
-- https://www.phpmyadmin.net/
--
-- Client : 127.0.0.1
-- Généré le : Mar 20 Juin 2017 à 13:22
-- Version du serveur : 5.7.14
-- Version de PHP : 7.0.10

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Base de données : `projet_lo07`
--

-----

--
-- Structure de la table `cursus`
--

CREATE TABLE `cursus` (
  `label` varchar(25) NOT NULL,
  `etudiant` int(6) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-----

--
-- Structure de la table `element_formation`
--

CREATE TABLE `element_formation` (
  `id` varchar(25) NOT NULL,
  `sem_seq` int(2) NOT NULL,
  `sem_label` varchar(6) NOT NULL,
```

```

`sige` varchar(6) NOT NULL,
`categorie` varchar(6) NOT NULL,
`affectation` varchar(6) NOT NULL,
`utt` varchar(1) NOT NULL,
`profil` varchar(1) NOT NULL,
`credit` int(2) NOT NULL,
`resultat` varchar(6) NOT NULL,
`cursus` varchar(50) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

-----

```

--
-- Structure de la table `etudiant`
--

```

```

CREATE TABLE `etudiant` (
  `id` int(6) NOT NULL,
  `nom` varchar(100) NOT NULL,
  `prenom` varchar(25) NOT NULL,
  `admission` varchar(4) NOT NULL,
  `filiere` varchar(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

-----

```

--
-- Structure de la table `regle`
--

```

```

CREATE TABLE `regle` (
  `id_regle` varchar(50) NOT NULL,
  `num_regle` int(4) NOT NULL,
  `action` varchar(10) NOT NULL,
  `type` varchar(20) NOT NULL,
  `temps_cursus` varchar(10) NOT NULL,
  `credits` int(4) NOT NULL,
  `idReglement` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

-----

```

--
-- Structure de la table `reglement`
--

```

```

CREATE TABLE `reglement` (

```



```
`nom_reglement` varchar(50) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
```

```
-- Index pour les tables exportées
```

```
--
```

```
--
```

```
-- Index pour la table `cursus`
```

```
--
```

```
ALTER TABLE `cursus`  
  ADD PRIMARY KEY (`label`),  
  ADD KEY `etudiant` (`etudiant`);
```

```
--
```

```
-- Index pour la table `element_formation`
```

```
--
```

```
ALTER TABLE `element_formation`  
  ADD PRIMARY KEY (`id`),  
  ADD KEY `sigle` (`sigle`),  
  ADD KEY `id` (`id`),  
  ADD KEY `cursus` (`cursus`);
```

```
--
```

```
-- Index pour la table `etudiant`
```

```
--
```

```
ALTER TABLE `etudiant`  
  ADD PRIMARY KEY (`id`);
```

```
--
```

```
-- Index pour la table `regle`
```

```
--
```

```
ALTER TABLE `regle`  
  ADD PRIMARY KEY (`id_regle`),  
  ADD KEY `num_regle` (`num_regle`),  
  ADD KEY `idReglement` (`idReglement`);
```

```
--
```

```
-- Index pour la table `reglement`
```

```
--
```

```
ALTER TABLE `reglement`  
  ADD PRIMARY KEY (`nom_reglement`);
```

```
--
```

```
-- Contraintes pour les tables exportées
```

```
--
```

```

--
-- Contraintes pour la table `cursus`
--
ALTER TABLE `cursus`
  ADD CONSTRAINT `fk_etu` FOREIGN KEY (`etudiant`) REFERENCES `etudiant` (`id`) ON
  DELETE CASCADE ON UPDATE CASCADE;

--
-- Contraintes pour la table `element_formation`
--
ALTER TABLE `element_formation`
  ADD CONSTRAINT `fk_cursus` FOREIGN KEY (`cursus`) REFERENCES `cursus` (`label`) ON
  DELETE CASCADE ON UPDATE CASCADE;

--
-- Contraintes pour la table `regle`
--
ALTER TABLE `regle`
  ADD CONSTRAINT `fk_reglement` FOREIGN KEY (`idReglement`) REFERENCES `reglement`
  (`nom_reglement`) ON DELETE CASCADE ON UPDATE CASCADE;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```