

Développement d'une application web de gestion des plaques d'immatriculations.

Curtis Martelet

Tuteur de stage :

Mr. Joseph Maillardet



Année 2019-2020

Sommaire

| | |
|---|----|
| Introduction..... | 3 |
| L'entreprise. | 4 |
| I. Présentation | 5 |
| 1. Introduction | 5 |
| 2. Les locaux. | 5 |
| 3. Partenariat. | 6 |
| 4. Emplacement. | 7 |
| 5. Structure juridique. | 8 |
| 6. Concurrence. | 9 |
| II. Histoire | 10 |
| 1. Chronologie de l'IUT. | 10 |
| 2. Direction de l'IUT. | 10 |
| Le stage..... | 11 |
| I. Cahier des charges..... | 11 |
| II. Le travail déjà existant..... | 11 |
| Travail réalisé..... | 12 |
| I. Introduction..... | 12 |
| II. La base de données. | 13 |
| 1. Conception de la base de données. | 13 |
| 2. Remplissage de la base de données. | 14 |
| III. Le PHP..... | 16 |
| 1. Connexion à la base de données | 16 |
| 2. Manipulation de la base de données..... | 19 |
| IV. Le HTML & CSS. | 22 |
| 1. HTML. | 22 |
| 2. CSS..... | 23 |
| Conclusion | 24 |
| I. Conclusion technique. | 24 |
| II. Conclusion personnelle. | 24 |
| Remerciement | 26 |
| Bibliographie & Annexe..... | 27 |

Introduction

L'intitulé de mon stage est « Développement d'une application web de gestion des plaques d'immatriculation ».

L'IUT n'a qu'une seule entrée, gardé par des agents de sécurités. Ces agents ne laissent rentrer que les personnes et véhicules ayant un lien avec l'IUT, c'est-à-dire les professeurs, le personnel et les étudiants. Leurs véhicules personnels, ainsi que les camions de livraisons ont aussi l'autorisation d'entrée. Pour que les gardiens différencient les véhicules faisant partie de l'IUT de ceux de l'extérieur, on colle sur les parebrises des véhicules un autocollant, que les agents doivent vérifier. Cette vérification doit se faire à l'arrêt, pour que les agents puissent correctement lire.

Chaque autocollant est lié à une fiche d'identification du véhicule, actuellement au format papier.

L'objectif de mon stage est de numériser ces fiches en permettant aux étudiants d'enregistrer eux-mêmes les données de leurs véhicules ainsi que leurs coordonnées de contacts. Une fois enregistrées, ces données seront validées par les agents de sécurités. Enfin, une interface web permettra de parcourir les données afin de retrouver le plus rapidement possible les données d'un véhicule. Par exemple, lorsqu'un véhicule est en stationnement gênant.

L'entreprise.



Image 1 : Façade de l'IUT

I. Présentation

1. Introduction

L'IUT de Nîmes est un établissement scolaire publique rattaché à l'académie de Montpellier, au même titre que la faculté de science ou l'IUT de Montpellier. Cet établissement dispense des cursus de DUT ainsi que des licences et formations professionnelles. Le campus de 5 hectares est constitué de 5 bâtiments, un pour chaque département :

- GEA, Gestion des Entreprises et des Administrations.
- GEII, Génie Electrique et Informatique Industriel.
- GC, Génie Civil.
- GMP, Génie Mécanique et Productique.
- SGM, Science et Génie des Matériaux.

C'est dans ces 5 départements que se répartissent les 1300 étudiants annuels¹. Majoritairement des étudiants ou des chercheurs d'emplois, ils viennent pour se former aux 5 DUT et 14 licences professionnelles que dispense l'IUT en cursus initial, continue ou alternance.

L'IUT accueille également une section de Polytech' Montpellier (*Formation d'Ingénieur, bac+5*) qui forme des ingénieurs aux systèmes chaudronnés et soudés.

Pour les encadrés, l'IUT peut compter sur les 53 membres du personnel, 97 professeurs présents sur le campus et 125 intervenants professionnels. ¹

L'IUT a formé en 50 ans plus de 20 000 étudiants. Avec un taux de poursuite d'étude de 85% et d'insertion professionnelles de 97%, Nîmes se place comme l'un des IUT de France avec le meilleur taux de réussite.

2. Les locaux.

L'IUT de Nîmes a investi dans de nombreuses infrastructures pour accompagner les étudiants dans leurs études. Sur le campus, on peut trouver :

¹ En 2016.

- 4 halles d'essais,
- Une plateforme d'analyse de microstructure, nommé Anamat,
- Une maison intelligente, qui permet de se former aux technologies des objets connectés,
- Une antenne du Centre Spatial Universitaire de Montpellier,
- Une plateforme pour l'apprentissage du développement des motorisations électriques,
- Une plateforme d'impression 3D, comportant une cinquantaine d'imprimante 3d en libre utilisation dans tout le campus.

L'IUT dispose aussi d'une bibliothèque universitaire équipé d'ordinateurs et d'imprimantes.

Membre du réseau Learning 'Lab, elle est équipée de 4 salles de type Learning' Lab. Ces salles, financés par la région et la ville, sont conçues pour aider les étudiants dans leurs projets scolaires et personnels.

Dans l'enceinte du campus se trouve un point chaud : petits déjeunés et fast-foods y sont servis tous les jours. Une alternative plus rapide et plus proche au restaurant universitaire, qui se trouve à 2 minutes à pied.

3. Partenariat.

En plus de 50 ans d'existence, l'IUT a tissé une relation privilégiée avec des acteurs privés.

C'est plus de 60 entreprises qui, chaque année, répondent présentes pour proposer aux étudiants stages, CDD et CDI. Une fois leur stage terminé, ces étudiants auront peut-être l'opportunité de continuer à travailler dans l'entreprise, en CDI ou CDD.

Chaque année, ces mêmes entreprises se donnent rendez-vous à l'IUT pour une journée de rencontre. Les étudiants ont la possibilité de rencontrer des entreprises, et ces entreprises profitent de ces rencontres pour former des partenariats.

Plus de 170 entreprises de la région finance les projets de formations mis en place par l'IUT. Ces aides permettront, par exemple, d'entretenir les infrastructures existantes ou d'investir dans de nouveaux projets.

L'antenne du Centre Spatial Université de Montpellier a permis à l'IUT d'ouvrir un laboratoire dédié au spatial. Ainsi, l'IUT a développé un nano satellite, appelé Robusta, qui a été mis en orbite en 2017. Un second satellite est actuellement en phase de développement.

Une licence professionnelle spécialisée dans les métiers du spatial a aussi vu le jour grâce à cette collaboration.

4. Emplacement.

Le site de l'IUT de Nîmes se situe rue Jules Raimu, dans le quartier de Saint-Césaire : c'est un quartier en périphérie de la ville.

On peut trouver à proximité une gare SNCF et des arrêts de transports en commun.

La cité universitaire de Nîmes se trouve à moins de 10 mn à pied : le CROUS de Nîmes est aussi dans ce bâtiment. Un chemin entièrement piéton a été aménagé entre les deux. Sinon, des chambres étudiantes peuvent être trouvées tout autour du site.

On peut aussi trouver à proximité de l'IUT plusieurs résidences où ils peuvent loger.

L'IUT se trouve à proximité de la zone industrielle de Nîmes. Il est donc facile pour les stagiaires de se retrouver, et facile pour les entreprises de venir à l'IUT.

L'IUT de Nîmes est donc très bien placé vis-à-vis du public qu'elle vise : les étudiants.

Loin de la ville, donc du vacarme et des embouteillages, elle est connectée au cœur de la ville par les transports en communs et plusieurs axes routiers.

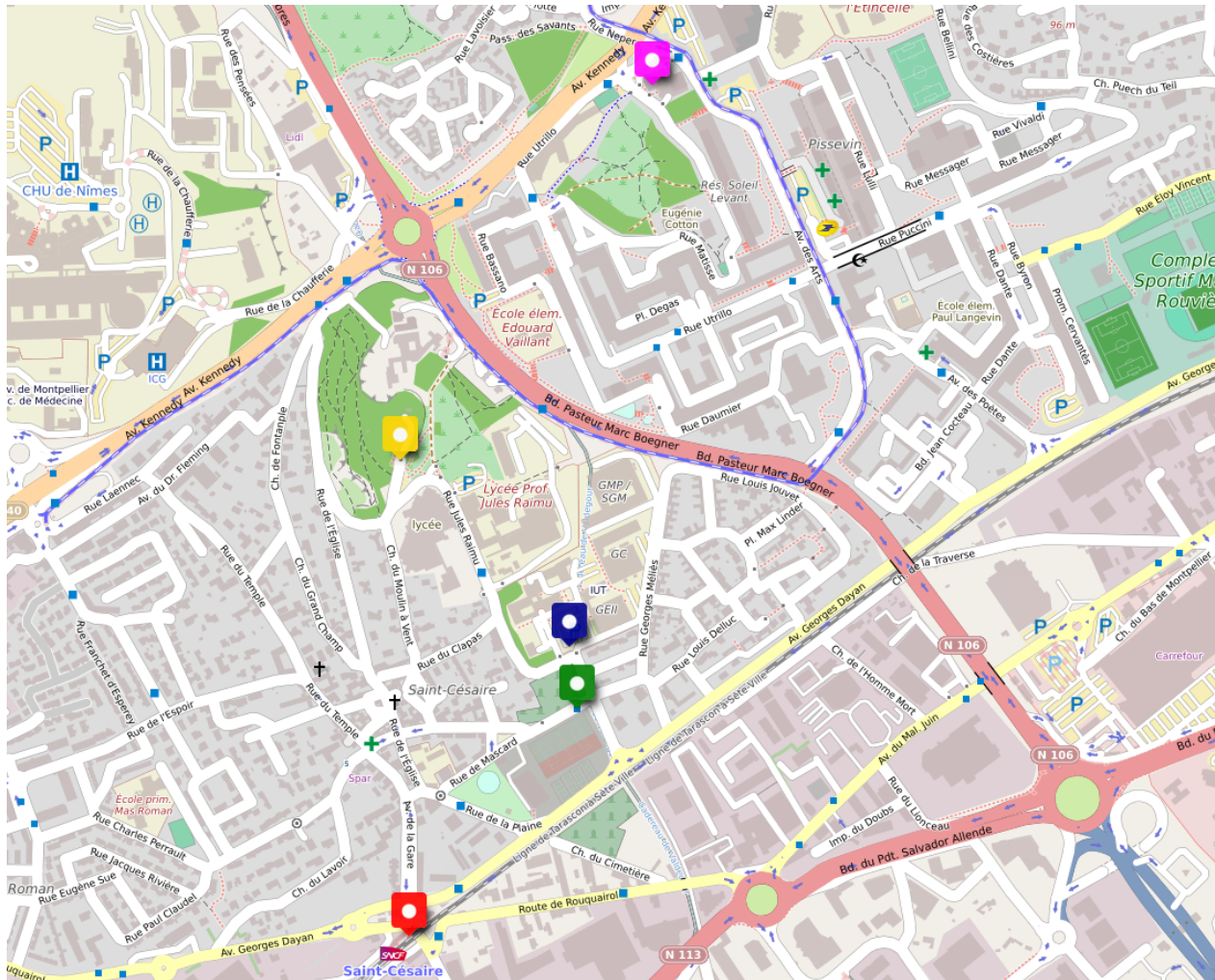


Image 2 : carte des environs de l'IUT

| Légende | | | | |
|--------------|--------------|--------------------------|-----------|--------------------|
| Bleu | Vert | Jaune | Rouge | Rose |
| IUT de Nîmes | Arrêt de bus | Restaurant universitaire | Gare SNCF | Cité universitaire |

5. Structure juridique.

L'IUT de Nîmes appartient depuis 2015 à l'université de Montpellier. Son statut juridique est donc celui d'une université. Il s'agit d'un « type particulier d'établissement public à caractère scientifique, culturel et professionnel » (EPSCPEPSCP). Les règles constitutives sont donc fixées par le livre 7 du code de l'éducation.

6. Concurrence.

Comme dit plus tôt, l'IUT de Nîmes est un établissement scolaire qui délivre des DUT et des licences professionnelles. Ses concurrents sont donc des établissements scolaires délivrant la même chose (même si l'on ne parle pas vraiment de concurrence). Je ne prendrais donc pas les écoles préparatoires ou l'université car, même si le niveau obligatoire (baccalauréat) est le même, le diplôme ne l'est pas.

Les DUT ne peuvent être délivrés que dans un IUT, or, il n'y a qu'un seul IUT à Nîmes : il a donc le monopole.

L'IUT de Montpellier², appartenant aussi à l'université de Montpellier, est spécialisé dans l'informatique : ils ne sont donc pas vraiment en concurrence.

Les licences professionnelles, à la différence des DUT, peuvent être dispensées par certaines universités. L'IUT de Nîmes dispose de licences professionnelles que ne propose pas l'université de Nîmes.

La seule concurrence que connaît l'IUT est avec des écoles privées. Elles peuvent délivrer un diplôme ayant un niveau d'étude équivalent à un DUT ou une licence professionnelle.

² J'ai choisi cet IUT car proche géographiquement et de par leur histoire commune.

II. Histoire

1. Chronologie de l'IUT.

L'histoire de l'IUT de Nîmes commence en 1968, lors de l'inauguration de l'IUT de Montpellier-Nîmes. Un an plus tard, le site de Nîmes est inauguré : il ne compte encore que 3 des 5 départements actuels.

En 1991 sera inauguré le département GEA, et en 1996, le département SGM.

En 1992, l'ensemble IUT Montpellier-Nîmes se sépare pour donner naissance à deux IUT : l'IUT de Montpellier d'un côté, et l'IUT de Nîmes de l'autre.

Avec la création des licences professionnelles en 2000, l'IUT a inclus plusieurs d'entre elles à son cycle d'enseignement.

En 2015, l'IUT a fusionné avec l'académie de Montpellier, devenant ainsi un composant de l'académie de Montpellier.

2. Direction de l'IUT.

De nombreuses personnes se sont succédées à la direction de l'IUT. Un organigramme de la direction actuelle se trouve dans [l'annexe](#).

Les décisions concernant la politique de l'IUT sont prises par le Conseil de l'IUT. Il formule aussi des propositions pour la mise en œuvre des politiques choisies.

Le président du conseil de l'IUT est élu pour une durée de trois ans, renouvelable, parmi des personnes extérieures représentant les milieux sociaux économiques. Le président est chargé d'assurer le bon fonctionnement du conseil.

Chaque département a son propre directeur. Il siège au conseil de l'IUT et est chargé de gérer le département qu'il représente.

Enfin, les divers services de l'IUT (restauration, communication, informatique...) peuvent aussi faire remonter au conseil des problèmes et requêtes.

Le stage

I. Cahier des charges.

On souhaite développer une application web qui permettra aux personnes membres du campus d'enregistrer leurs données par l'intermédiaire d'une interface simple et accessible sur téléphone mobile.

Ces informations ne pourront être modifiable que par l'étudiant, et visionnable que par ceux ayant eu l'autorisation de la direction (agent de sécurité, direction...). Enfin, n'importe quel utilisateur devra pouvoir être contacté par le biais de son téléphone portable ou par email.

Ce projet est conséquent. Il nécessite que l'on construise une base de données et un site web doté d'une interface au design adaptable à tous les écrans (ordinateur, téléphone...), on parle alors de « responsive design ». Il faudra, dans l'ensemble, pouvoir se connecter à un site, ajouter ou modifier les données présentes dans une base de données à partir du site. Enfin, il faudra afficher les données que l'on voudra à partir du site...

II. Le travail déjà existant.

Un prototype du site web m'a été fourni au début du stage.

Au cours de celui-ci, mon tuteur m'a aussi transmis des documents liés au site et à la base de données. Ces documents sont : le schéma de la base de données et un début de script SQL.

Travail réalisé

I. Introduction.

Le projet est, comme dit plus haut, énorme.

Dans un premier temps, j'ai récapitulé tout ce qui devait être fait dans ce projet. Une fois fait, j'ai testé les différentes tâches à faire, pour trouver ce qui m'intéressait le plus. J'en ai conclu que j'allais me concentrer sur la partie base de données du projet : la structure du site étant bien avancé, j'ai préféré me concentrer sur ce qui n'avait pas été commencé.

J'ai réalisé le script qui préparera la base de données, les fonctions qui connecteront cette base au site et celles qui permettront la manipulation de la base de données par du PHP.

J'ai réalisé mon stage en télétravail : j'ai dû travailler de chez moi.

Cette organisation m'a obligé à revoir mon environnement de travail. Aidé par mon tuteur, j'ai installé sur mon PC les programmes et logiciels nécessaires à mon stage. Entre autres, j'ai ajouté à mon PC une machine virtuelle contenant un serveur local. Cependant, cette installation n'a pas été sans problème puisque mon PC a crashé un bon nombre de fois durant mon stage, ralentissant mon travail (en supprimant les dernières modifications apportées à un code par exemple). Le logiciel de gestion de base de données que mon tuteur m'a conseillé est MariaDB.



*Image 3 : Logo de MariaDB,
logiciel de gestion de base de données.*

II. La base de données.

1. Conception de la base de données.

Avant de la programmer, une base de données se réfléchit. Pour ce faire, des méthodes ont été développées.

Dans le cadre de ce projet, mon tuteur m'a initié à un certain nombre de méthodes, dont la méthode Merise.

La méthode Merise est une méthode mise en place dans les années 70 pour aider à l'informatisation des entreprises. Méthode européenne, elle a connu plusieurs évolutions, avec Merise/objet et Merise/2 (j'ai utilisé cette dernière).

Cette méthode est adaptée pour les projets internes à une organisation. Elle est facile à comprendre, donc idéale pour un débutant comme moi.

Je me suis quand même intéressé aux autres, mais seule celle-ci avait une documentation, vidéos et tutoriels compréhensibles.

La méthode extraite de Merise se découpe en deux parties.

1. Dans un premier temps, on réfléchit aux tables et variables qui devront être dans la base de données. On identifie quelles variables pourraient être remplies plusieurs fois, et la dépendance qu'elles ont avec les autres variables. Ainsi (par exemple), la table email a été séparée de la table personne car il est possible qu'il ait plusieurs adresses mails ; a été également séparée la table coordonnée car il est aussi possible que l'utilisateur ait un seul mail mais plusieurs adresses, ou inversement. Cette méthode permettra à l'étudiant de mettre autant d'information qu'il le souhaitera, en la remplissant le moins possible le stockage du serveur.

2. Dans un second temps, on s'intéresse aux associations. On identifie quelle table est dépendante de qui. On utilise la clé principale de chaque table, nommé id, pour faire ces associations : on recopie l'id d'une table vers l'autre pour les associer.

Cette étape fut plus dure que la précédente. Je n'arrivais pas à faire une association convenable entre l'utilisateur et les formations/départements.

Après quelques jours de recherches, mon tuteur m'a partagé les résultats de ses recherches après lui avoir expliqué mes trouvailles. Son schéma était similaire au miens, je m'en suis donc inspiré. Il se trouve en [annexe](#).

La partie conception terminée, il était temps d'écrire le script.

2. [Remplissage de la base de données.](#)

Le remplissage de la base de données se fera à l'aide d'un script écrit en SQL, le langage normalisé des bases de données rationnelle.

Comme me l'a expliqué mon tuteur, il est préférable d'écrire un script pour le remplissage de la base de données. Ainsi, si la base de données est supprimée, on pourra facilement la recréer.

Ce premier script a trois objectifs : créer les tables, faire les associations et injecter les données préliminaires nécessaires à son bon fonctionnement.

Hormis les associations qui prirent un peu trop de temps (car je ne les avais pas bien comprises), ce script fut simple à écrire.

Il a aussi d'abord définir les types de données qui seront présents dans chaque colonne de chaque table. J'ai rencontré un problème mineur avec la clé principale. Je voulais utiliser le paramètre **SERIAL**, qui est conçu pour ce type de données, mais dans mon cas, ce paramètre faisait planter ma base de données. J'ai donc utilisé une série de paramètre équivalent à SERIAL.

***SERIAL** défini les valeurs comme étant des chiffres strictement positifs,
s'incrémentant automatiquement.*

*Je l'ai remplacé de sorte de définir ces variables comme des chiffres (**INT**)
strictement positif (**UNSIGNED**), non nul (**NOT NULL**), et s'incrémentant
automatiquement (**AUTO_INCREMENT**).*

L'ordre de création des tables à son importance : on ne peut pas faire de référence à une table qui n'existe pas. J'ai donc trouvé l'ordre qui permet de créer les tables, en priorisant celles qui n'ont pas de référence, et en finissant par celle en ayant le plus.

Enfin, j'ai écrit les fonctions qui prérempliront les tables qui doivent l'être.

Comme lors de la conception de la base de données, mon tuteur m'envoya son script quelques jours plus tard. J'en ai récupéré certaines choses, notamment la méthode de création des associations, que j'ai ensuite mis dans mon script. Ses associations utilisent une fonction qui permet de les nommer, ce qui permet de les appeler pour les modifier, les supprimer ou les visualiser.

```
1. CREATE TABLE `Telephone` (  
2.   `id_telephone` SERIAL PRIMARY KEY,  
3.   `numero` INT(10),  
4.   `acronyme` VARCHAR(100),  
5.   `id_personne` INT UNSIGNED NOT NULL,  
6.  
7.   CONSTRAINT `cont_telephone_idpersonne`  
8.     FOREIGN KEY (id_personne) REFERENCES Personne (id_personne)  
9. );  
10.
```

Code 1 : Script pour créer la table Telephone

La ligne 7 et 8 sont une illustration de la méthode employée par le tuteur.

Une fois la table remplie et opérationnelle, il est temps de passer au PHP.

III. Le PHP.

Le PHP est un langage beaucoup plus difficile que tous ceux que j'ai croisé jusqu'à présent (C, C++, SQL).

Jusqu'à présent, je n'ai utilisé que des langages ayant une structure impérative, c'est-à-dire que l'on écrit son code sur une dimension (les fonctions s'exécutent à la suite). Le PHP est un langage orienté objet, ce qui fait que la structure du code est fondamentalement différente. J'ai donc dû réapprendre la programmation (à un certain niveau).

Comme conseillé par mon tuteur, j'ai réalisé des projets personnels pour me familiariser avec cette nouvelle construction. J'ai fait en sorte que tous aient un lien avec mon stage. J'ai ainsi, par exemple, recréer la base de données à l'aide de classe, classe que j'ai par la suite remplie et affichée.

Le début de site que mon tuteur m'a envoyé m'a permis de comprendre le fonctionnement et l'utilisation du PHP. J'ai ainsi pu m'inspirer de la structure et de certaines classes pour mon propre programme.

1. Connexion à la base de données

Une fois le langage en main (ce qui ne me prit deux ou trois semaines de travail), j'ai commencé la rédaction de la fonction qui connectera le site à la base de données. Cette étape fut la plus longue : elle me prit un bon mois.

PHP ne peut pas piloter une base de données tout seul : il a besoin de pilotes et d'extensions. Ces pilotes changent en fonction du logiciel de gestion.

Rappel : le logiciel de gestion que j'utilise est MariaDB.

Dans un premier temps, j'ai appris à utiliser MySQLi, un pilote propriétaire du logiciel MySQL. Puisque MariaDB est un fork de MySQL, ce pilote fonctionne, mais ce ne sera très certainement plus le cas d'ici quelques années. Mon programme fonctionnant, mon tuteur m'a demandé de passer sur PDO.

PDO peut être utilisée avec tous les logiciels de base de données (MySQL, MariaDB, Mongo DB...).

Son utilisation est plus simple que MySQLi, nécessitant moins de requête pour faire la même chose.

Un des avantages de PDO est le nombre de collision : beaucoup moins important qu'avec MySQLi. Les collisions désignent le risque que des données de deux utilisateurs se mélangent si elles sont écrites en même temps dans la même table.

J'ai donc dû recommencer mon programme pour intégrer l'utilisation de PDO. C'est à partir de ce moment-là que les problèmes ont commencé.

PDO, à la différence de MySQLi, a beaucoup moins de fonctions. Il ne dispose pas de fonctions de débogage, notamment celles permettant de vérifier si une connexion est faite, ou de savoir la raison de la perte de connexion. Je n'ai donc pas tout de suite compris que le serveur qui hébergeait ma base de données empêchait le navigateur de s'y connecter.

Pour résoudre ce problème, j'ai essayé toutes les méthodes que j'ai pu trouver : inonder ma base de données de requête, activer la persistance de la connexion, modifier les paramètres de mon logiciel... mais rien n'y faisait.

J'ai suivi les conseils d'un ami informaticien qui, pour se connecter à ses bases de données, utilise une fonction : try/catch.

Cette fonction ne marche qu'avec des exceptions. C'est pourquoi on doit modifier le comportement en cas d'erreur (de base, si la base de données envoie une erreur, PDO ne fait rien), en le définissant sur **ERRMODE_EXCEPTION** : on oblige le PDO à donner l'erreur sous forme d'exception. Cette exception est alors affichée.

```

1.     try{
2.         $pdo=new PDO("$driver:host=$host;dbname=$db",$username,$password
, array(
3.             PDO::ATTR_ERRMODE=>PDO::ERRMODE_EXCEPTION
4.         ));
5.         return $pdo;
6.     } catch(PDOException $e){
7.         echo 'Échec lors de la connexion : ' . $e->getMessage();
8.     }

```

Code 2 : Fonction permettant la connexion du navigateur.

Dans le code ci-dessus, on tente de se connecter à la ligne 2. Si une erreur a lieu, l'exception de PDO, nommé PDOException, est interceptée et stocké dans \$e. L'écho contenu dans le catch l'affiche ensuite.

Cette fonction pourra être amélioré, en faisant en sorte que l'erreur ouvre une page web qui contient une explication de l'erreur et de l'assistance (par exemple).

Après avoir appliqué la méthode de mon ami, j'ai enfin compris le problème.

Le problème était causé par le logiciel de gestion de la base de données : mon interface de manipulation de base de données m'indiquait que MariaDB était installé, alors que l'erreur m'expliquait que j'avais en fait MySQL sur mon PC. J'ai alors modifié mon programme pour prendre en compte ce changement. Suite à ce changement, tout fonctionne.

Pour faciliter le travail du développeur qui me suivra, j'ai modifié mon programme pour utiliser un fichier de configuration (extension .ini). Ce fichier permet de modifier les paramètres de connexions sans avoir à se plonger dans le programme. La fonction *parse_ini_file* stock toutes les données du fichier dans un tableau.

```

1. $connexion = is_file("$file.ini") ? parse_ini_file("$file.ini") : false;
2.
3.     $driver=$connexion['driver'];
4.     $db=$connexion['database'];
5.     $host=$connexion['host'];
6.     $username=$connexion['username'];
7.     $password=$connexion['password'];

```

Code 3 : Utilisation de parse_ini_file

Si le fichier \$file.ini existe, on stock son contenu dans \$connexion, sinon, on renvoie false (faux).

J'ai par ailleurs travaillé sur une fonction qui permet de savoir si l'on est encore connecté à la base de données. A chaque appel, on tentera de lire une valeur précise. Si la fonction renvoie

TRUE, alors on est bien connecté, sinon, on ne l'est pas.

Mais par manque de temps, je l'ai laissé de côté puisque utile que dans la phase de débogage.

2. Manipulation de la base de données.

La connexion réalisée, j'ai commencé la 3^e et dernière partie de mon projet : écrire la partie qui commande. Cependant, la précédente étape a utilisé beaucoup trop de temps, ce qui fait que je ne l'ai pas testé.

Cette partie comporta deux étapes. La première consiste à écrire les scripts SQL que l'on exécutera. La seconde fera en sorte que le programme PHP complète les scripts, et les envoie. Due au retard que j'ai accumulé à cause de la seconde partie, je n'ai pas pu terminer cette seconde étape.

La première partie fut assez rapide. J'ai réfléchi puis écrit en SQL les scripts qui permettront au PHP de :

- Supprimer une table,
- Modifier les données d'une table,
- Ecrire dans une table,
- Lire une table,
- Purger la base de données.

Ces scripts contiennent des paramètres. Ils seront modifiés en fonction de ce que l'on veut manipuler. Pour plus de clarté, j'ai ajouté à ces paramètres un `_` pour les identifier.

La seconde étape fut bien plus difficile à écrire. Le plus gros obstacle de cette partie réside dans les deux fonctions que je n'avais jamais vues : `prepare` et `execute`.

Ces fonctions prépare des scripts SQL, puis les exécute plus tard dans des fonctions PHP. Les scripts préparés peuvent être modifiés au moment de leur exécution. Cette méthode fait sens si l'on a à l'esprit que ces scripts seront appelés plus d'une fois par connexion. Ainsi, en gardant la

structure mais en modifiant les paramètres, on peut faire les opérations que l'on veut, sans avoir à les réécrire. Cela diminue le volume de code à écrire, le risque d'erreur lors de l'écriture, et les tentatives de piratage.

```
1. $sql4 = "SELECT `:colonne` FROM `:table` WHERE `:condition`";
2. $afficher = $pdo->prepare($sql4);
3.
4. $afficher->execute('/:colonne',"id",':table',"identifiant",':condition',"$condition");
```

Code 4: Exemple utilisation de prepare/execute.

Dans l'exemple ci-dessus, \$sql4 stock un script qui a vocation de récupérer/afficher des valeurs d'une table. On prépare le script en le mettant dans la variable \$afficher. Au moment de l'exécution, on personnalise le script pour qu'il récupère l'id de la personne qui correspond à \$condition.

Je n'ai pas eu le temps de finir les protections contre les tentatives de modifications interdites.

Il est possible que des hackers tentent de trafiquer la base de données à l'aide de techniques de piratage. La plus connue étant l'injections SQL.

Ces injections sont des scripts SQL qui sont écrits dans les formulaires HTML. Ils sont ensuite envoyés dans le logiciel de base de données, qui les interprète.

Ces scripts peuvent modifier les données, voir purger la base de données. Mais leur usage le plus problématique est de créer un utilisateur ayant des privilèges d'administrateur : une personne non autorisée aurait ainsi accès à toutes les données de la base de données, alors qu'il n'en a pas l'autorisation.

Pour que cela soit plus clair, voilà un exemple :

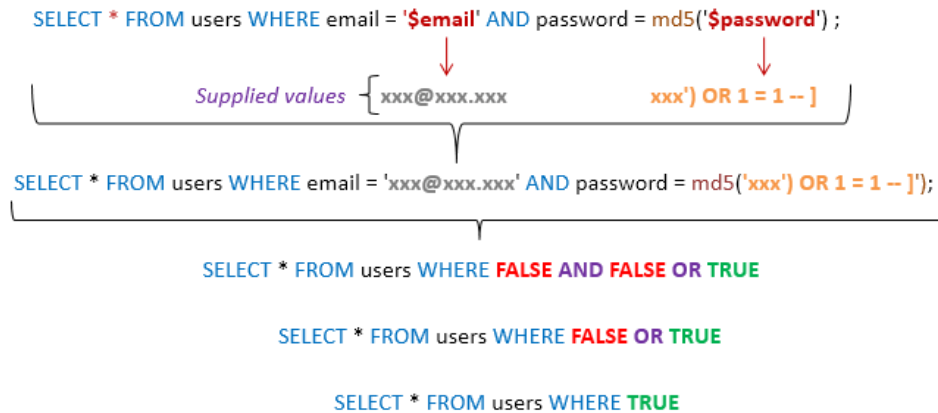


Image 4 : Injection SQL qui affiche l'intégralité de la table USER

Dans l'exemple ci-dessus, l'injection permet d'afficher toutes les informations de la table USERS : le pirate aura accès à tous les emails et mots de passe contenu dans la table. Si la table contient plus de données, alors il pourra aussi y avoir accès.

Pour s'en protéger, plusieurs méthodes existent. Dans mon programme, j'utilise la méthode la plus simple. Cette méthode fait en sorte que les variables soient lues comme étant strictement du texte. Ainsi, les tentatives seront comprises comme étant du texte.

Une autre méthode serait d'empêcher la saisie de certains caractères, tel que les %, ` / ou \. Cependant, ces méthodes peuvent être facilement contrées, c'est pourquoi j'avais prévu de créer des procédures qui seront stockées dans la base de données. Ces procédures rendent les tentatives de piratage quasi-impossibles.

IV. Le HTML & CSS.

J'ai passé la majorité de mon temps à travailler sur la base de données, en écrivant en SQL et en PHP. Cela ne m'a pas empêché d'écrire en HTML et en CSS.

1. HTML.

Le HTML est le squelette d'un site web. C'est lui qui permet de faire des formulaires et des liens hypertextes (lien d'une page à une autre). Il est aussi utilisé pour baliser le code, qui sera ensuite mis en page par le navigateur.

Dans un premier temps, j'ai expérimenté avec la première version du site que mon tuteur m'a transmis. J'ai tenté de modifier son code pour créer les éléments qu'il manquait : les formulaires. Je n'y suis pas arrivé : le programme HTML est enchainé à des fonctions PHP très difficile à comprendre pour un débutant comme moi. J'ai cependant réfléchi à tous les formulaires dont on aura besoin.

Dans un second temps, je me suis servi du HTML pour agencer ma page de débogage. Pour ne pas me perdre dans tous les messages d'erreurs/validation d'opération, j'ai fait en sorte que chaque application/fonctions soient séparées et titrées.

Mais mon utilisation principale de l'HTML fut la création de formulaires. On les utilise pour récupérer des données, et les modifier. Ces données sont ensuite récupérées grâce au PHP puis envoyé dans la base de données avec le SQL.

Le HTML ne m'a pas posé beaucoup de problèmes de compréhension. Excepté dans un cas de figure, avec le premier formulaire que j'ai créé.

Un formulaire peut employer deux méthodes pour envoyer des informations : les méthodes POST et GET.

GET passe par l'URL pour envoyer des informations. Cette méthode a comme avantage de pouvoir partager une page qui est déjà préremplie (puisque l'on envoie aux autres l'URL, qui contient les données préremplies).

POST est cependant plus sécurisé et fiable que GET, puisque les informations circulent dans une table interne.

Mes premiers formulaires ont été écrits avec GET, pour pouvoir lire dans l'URL les données envoyées. Cependant, il arrivait très souvent que rien ne soit envoyé, ou qu'une chaîne soit modifiée lors de l'envoi. Dans les deux cas, des erreurs apparaissaient.

Suivant les conseils de mon tuteur, j'ai converti mes formulaires en POST, ce qui a réglé le problème, mais a compliqué le débogage.

2. CSS

Le CSS est un langage complémentaire au HTML qui permet de créer et éditer le style d'un site web. Comme un traitement de texte, on peut choisir de colorier, barrer, agrandir (etc...) le texte contenu dans des balises HTML.

J'ai seulement utilisé le CSS pour styliser les formulaires et les titres de ma page. Mon but était uniquement de comprendre le langage et voir ses utilisations possibles une fois le site fini.

Conclusion

I. Conclusion technique.

Vu l'ampleur de ce projet et mes connaissances initiales en programmation orienté objet, il m'était impossible de le finir en seulement 2 mois. C'est pourquoi je me suis fixés plusieurs objectifs réalisables.

Cependant, je n'ai pas réussi à tous les atteindre : je pense que dans un contexte autre que celui d'une pandémie, j'aurais eu le temps.

Le confinement a rendu ce stage plus dur que ce que je pensais. Je devais contacter mon tuteur, qui avait son propre emploi du temps et qui, de par son métier, était lui-même confronté à de nombreux problèmes résultant du confinement. Cela fait que j'ai passé beaucoup de temps sur les forums et sur les référentiels techniques pour me documenter. J'ai dû apprendre de 0 tous les langages que j'ai utilisés, ce qui m'a ralenti.

Néanmoins, mon travail sera utile pour mes successeurs. J'ai commenté toutes les fonctions inachevées de mon programme, et facilité l'utilisation de celles que j'ai terminées.

Le projet est loin d'être fini. Certaines de mes fonctions peuvent encore être améliorées, et tout ce dont je ne me suis pas occupé, c'est-à-dire le site web, installer sur les serveurs de l'IUT le site, restent encore à faire.

II. Conclusion personnelle.

Ce stage m'a permis de m'occuper durant ces longs mois d'inactivités. L'ambiance générale due au confinement et à la pandémie commençait à me peser, et travailler sur un projet qui m'a passionné m'a permis de concentrer sur quelque chose.

J'ai programmé, une activité que j'adore, avec une personne qui avait déjà plusieurs années d'expériences. Le télétravail fut une expérience très enrichissante, me forçant à travailler en autonomie, et avec mes propres ressources. Cette expérience m'a appris beaucoup de chose en informatique, et sur les méthodes de travail.

Enfin, ce stage m'a permis de découvrir plusieurs métiers que je ne connaissais pas ou peu : administrateur de base de données, concepteur web... J'ai repensé mes études. Je vais

beaucoup plus m'intéressé aux voies offertes par l'informatique, et si possible, me diriger vers une carrière en informatique plus qu'en électronique.

Remerciement

Je voudrais remercier mon tuteur de stage, Monsieur **Joseph Maillardet**, pour m'avoir pris en stage malgré le contexte de pandémie.

Bibliographie & Annexe

Information IUT :

<https://iut-nimes.edu.umontpellier.fr/liut/presentation/>

<https://www.umontpellier.fr/universite/composantes/iut-nimes>

<http://www.onisep.fr/Ressources/Univers-Postbac/Postbac/Occitanie/Gard/IUT-de-Nimes-Universite-de-Montpellier>

https://fr.wikipedia.org/wiki/Institut_universitaire_de_technologie_de_N%C3%AEmes

<https://iut-nimes.edu.umontpellier.fr/files/2019/04/dossier-de-presse-50-ans-IUT-de-Nimes.pdf>

Information sur le PHP :

<https://www.php.net/manual/fr/pdo.prepare.php>

<https://www.php.net/manual/fr/pdostatement.execute.php>

Méthode merise :

[https://fr.wikipedia.org/wiki/Merise_\(informatique\)](https://fr.wikipedia.org/wiki/Merise_(informatique))

<https://www.youtube.com/watch?v=1zPUqXADuYE>

Illustration injection SQL :

<https://www.guru99.com/learn-sql-injection-with-practical-example.html>

Documents annexes :

Page 28 : Organigramme de l'IUT.

Page 29 : Schéma de la base de données.