

Projet GM1 :

Développement d'une plateforme en ligne pour le service
prospection EISTI



Réalisé par :

EL KASMI Adnane GM2 (CPGE) :	<code>elkasmiadn@eisti.eu</code>
FRANTZ Charles-Philippe GM2 (CPI) :	<code>frantzchar@eisti.eu</code>
KOUROUMA Aboubacar GM4 (CPGE) :	<code>kouroumaab@eisti.eu</code>
Vo Christelle GM4 (CPI) :	<code>vohoangmai@eisti.eu</code>

22 Décembre 2019

Table des matières

I	Introduction	2
II	Mode d'emploi	3
III	Tâches et obligations	8
III.1	Répartition des tâches	8
III.2	Exigences	8
IV	Cahier des charges	9
V	Organisation	10
V.1	Controllers	10
V.2	Index.php	10
V.3	Home.php	10
V.4	DataBase	10
V.5	JS	12
V.6	CSS	12
V.7	Html2Pdf	12
V.8	Uploads	12
V.9	Cahier_des_charges	12
V.10	Les images	12
V.11	Views	12
V.11.1	Administrateurs	12
V.11.2	Elèves	13
V.11.3	Autres	14
VI	Ce que nous avons réussi	15
VII	Problèmes rencontrés	16
VII.1	Problèmes résolus	16
VII.2	Problèmes en cours	16
VIII	Axes d'améliorations	17
IX	Conclusion	18

I Introduction

Dans le cadre du module « Développement web », nous devons réaliser un site en groupe qui mêle des initiés (anciens CPI) et des débutants (anciens CPGE). Notre groupe se compose de quatre étudiants en ING1-GM : Adnane EL KASMI GM2 (CPGE), Charles-Philippe FRANTZ GM2 (CPI), Aboubacar KOUROUMA GM4 (CPGE), et Christelle Vo GM4 (CPI).

Ce projet a pour objectif de mettre en pratique les langages étudiés en développement web. Le but de ce rapport est de présenter le projet en s'adaptant aux lecteurs. En effet, le rapport doit être compréhensible à la fois pour un professeur en programmation, pour un intervenant en relations humaines, et pour les représentants des clients (ici, le service prospection).

Parmi les sujets proposés par les associations étudiantes Cap'Eisti et Atila, nous avons choisi le développement d'une plateforme en ligne pour le service prospection. En effet, le système actuel de gestion de la prospection se base essentiellement sur un partage de documents dont le principal est un fichier Excel affichant les informations d'une prospection à chaque ligne. Les principales limites de ce fichier sont que nous ne pouvons ni filtrer (par type de prospection par exemple) ni ordonner (dans l'ordre du code postal par exemple) le tableau comme chacun le souhaite, que les étudiants ne peuvent pas s'inscrire directement en ligne et, par conséquent, il y a un décalage entre le moment où un étudiant vient confirmer sa participation dans le bureau du service prospection et le moment où le nombre de places est mise à jour dans le fichier. Parmi les problèmes qui en découlent, un étudiant peut venir au bureau pour demander à participer à une prospection puis, après confirmation orale du service prospection, un autre étudiant vient dans la journée pour demander à participer à la même prospection. De plus, l'inscription directe sur une plateforme en ligne permet d'éviter au service prospection d'être impliqué dans des négociations telles que l'échange de places entre deux étudiants.

Nous avons choisi ce sujet car la demande du client (service prospection) nous a semblé claire et réalisable par rapport à notre niveau. En effet, nous savions précisément ce que le client attendait de cette plateforme, sachant que nous avons nous-mêmes expérimenté le système de partage de documents actuel, donc nous avons rapidement cerné les fonctionnalités essentielles et les améliorations à effectuer. Plus tard, après réception du cahier des charges détaillé rédigé par le service prospection lui-même, nous visualisons facilement la structuration du site web demandé, ce qui aidera à l'organisation du travail.

Afin de rendre compte notre travail, nous rédigerons un mode d'emploi pour afficher correctement notre site web, puis nous relèverons nos tâches et obligations, ensuite nous montrerons comment nous avons répondu au cahier des charges, puis nous expliquerons l'organisation du projet, ensuite nous énumérerons ce que nous avons réussi, ensuite nous parlerons des problèmes rencontrés et, enfin, nous proposerons des axes d'améliorations pour notre site web.

II Mode d'emploi

Afin de visualiser le fruit de notre travail, nous avons écrit un mode d'emploi pour faire fonctionner notre programme, et donc accéder à notre site web.

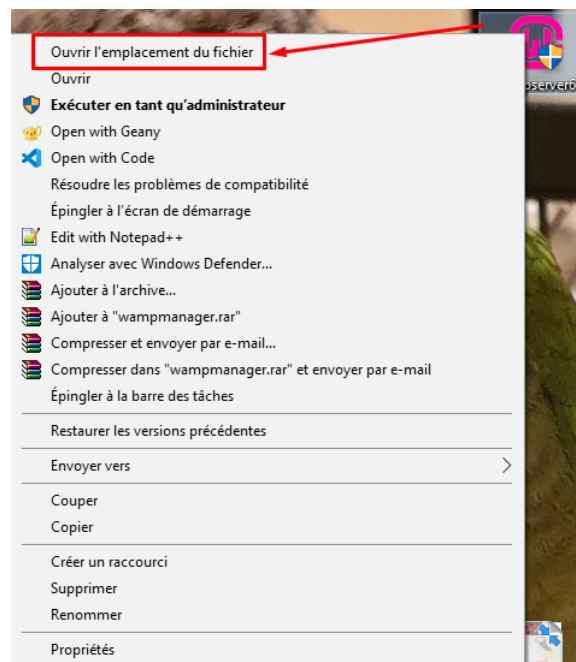


```
1 <?php
2 ob_start();
3 error_reporting(0);
4
5 // Constants
6 define("USERNAME", "root");
7 define("PASSWORD", "");
8 define("DSN", "mysql:host=localhost;dbname=test;charset=utf8");
9
```

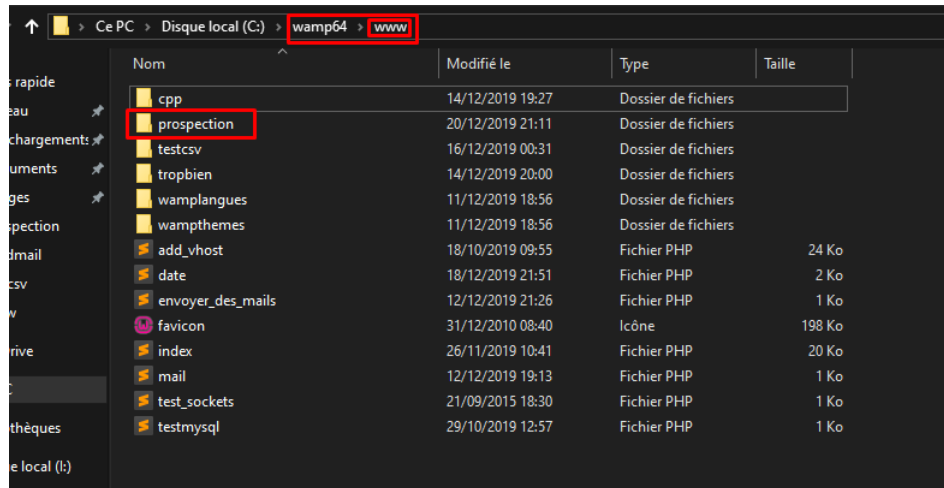
FIGURE 1 – Portion de code à respecter dans le fichier *index.php*

Pour faire fonctionner notre programme sous le système d'exploitation Windows, il faut respecter le nom de la database dans le champ *dbname* se situant dans le fichier *index.php*. Ici, le nom de la database à utiliser dans *phpMyAdmin* est **test**.

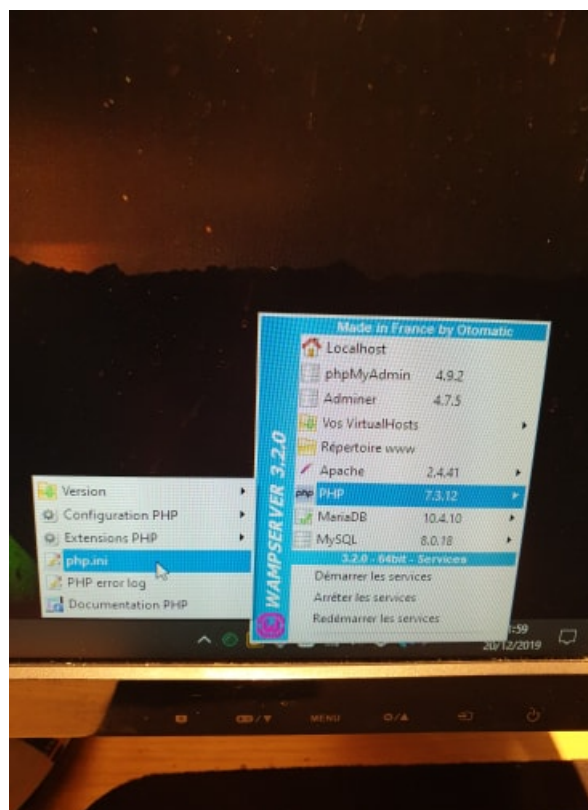
Pour ce qui est de la base de données sous Windows, il faut installer et lancer WAMP.



Etape 1 : Il faut installer, puis aller dans le répertoire où est installé WAMP. Le moyen le plus rapide est d'aller sur le raccourci se trouvant sur le bureau, puis de faire un click droit, et de cliquer sur « Ouvrir l'emplacement du fichier », comme vous pouvez le voir sur la figure ci-dessus.



Etape 2 : Une fois arrivé dans le dossier de WAMP, il faut aller dans le dossier *www*, et y placer notre dossier du projet, comme vous pouvez le voir sur la figure ci-dessus.



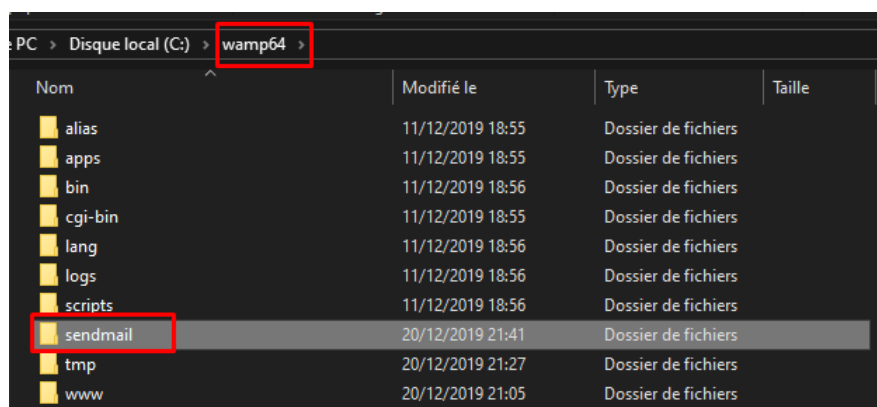
Etape 3 : Afin de configurer le système d'envoi de mails automatiques, il faut modifier quelques lignes du fichier *php.ini* de WAMP. Pour y accéder, il faut faire un clic gauche sur l'icône de WAMP en bas à droite de l'écran et aller dans PHP, puis faire un clic gauche sur *php.ini*. Une fenêtre avec le contenu du fichier *php.ini* va s'ouvrir.

```

1064
1065 ;phar.cache_list =
1066
1067 [mail function]
1068 ; For Win32 only.
1069 ; http://php.net/smtp
1070 SMTP = smtp.it-connect.fr
1071 ; http://php.net/smtp-port
1072 smtp_port = 25
1073
1074 ; For Win32 only.
1075 ; http://php.net/sendmail-from
1076 sendmail_from = "php@it-connect.fr"
1077
1078 ; For Unix only. You may supply arguments as well (default: "sendmail -t -i").
1079 ; http://php.net/sendmail-path
1080 sendmail_path = "C:\wamp64\sendmail\sendmail.exe"
1081
1082 ; Force the addition of the specified parameters to be passed as extra parameters

```

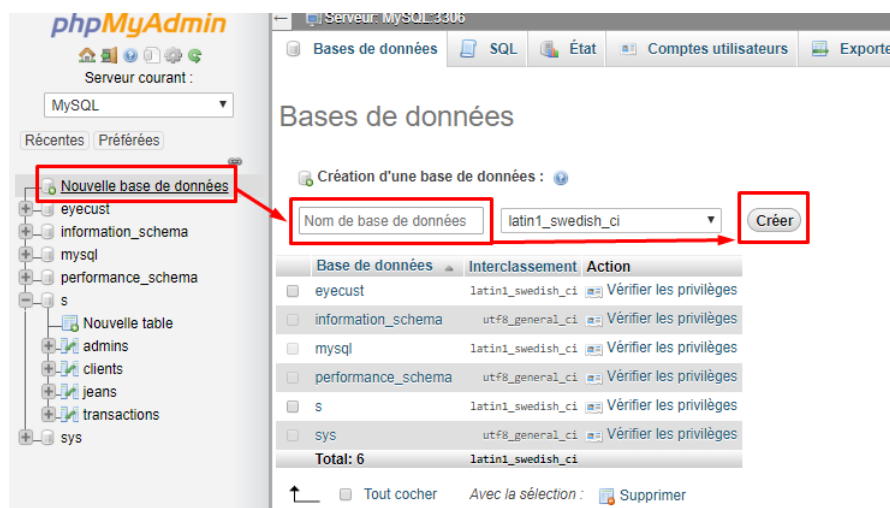
Etape 4 : Une fois le fichier *php.ini* ouvert, il faut maintenant chercher les lignes qui nous intéressent. Pour cela, effectuer un CTRL+H et chercher le mot clé « sendmail ». Les lignes encadrées ci-dessus en rouge doivent être les mêmes que dans votre fichier *php.ini* mais, attention, pour la ligne du **sendmail_path**, il faut bien voir où votre dossier WAMP est installé, voir s'il s'appelle Wamp ou wamp64 comme sur la photo, et choisir le bon disque dur (ici C :).



Etape 5 : Le dossier **sendmail** contenant les différents fichiers pour pouvoir effectuer les envois de mails doit se placer dans le dossier wamp64 ou Wamp selon le nom de votre dossier WAMP. Pour ce projet, les mails automatiques seront envoyés à partir du compte gmail **service.prospection.eisti@gmail.com** avec comme mot de passe **prospection95**.



Etape 6 : Il faut aller sur *phpMyAdmin* en exécutant l'adresse *localhost/phpmyadmin* afin d'arriver à l'affichage ci-dessus. Le nom d'utilisateur que nous utilisons est « root », et le mot de passe est vide. Ensuite, il faut appuyer sur le bouton « Exécuter » afin d'accéder à la base de données.



Etape 7 : Une fois connecté dans *phpMyAdmin*, il va falloir créer une nouvelle base de données. Pour ce faire, il faut appuyer sur « Nouvelle base de données », puis lui donner un nom (pour ne pas avoir de conflits avec notre projet, veuillez appeler votre base de données avec le nom **test**), et appuyer sur le bouton « Créer ».

Structure SQL Rechercher Requête Exporter Importer

Importation dans la base de données « test »

Fichier à importer :

Le fichier peut être compressé (gzip, bzip2, zip) ou non.
Le nom du fichier compressé doit se terminer par `.[format].[compression]`. Exemple : `.sql.zip`

Parcourir les fichiers : **Choisir un fichier** Aucun fichier choisi (Taille maximale : 128Mio)

Il est également possible de glisser-déposer un fichier sur n'importe quelle page.

Jeu de caractères du fichier : utf-8

Importation partielle :

☒ Permettre l'interruption de l'importation si la limite de temps configurée dans PHP est sur le point

Ignorer ce nombre de requêtes (pour SQL), à partir du début : 0

Autres options :

☒ Activer la vérification des clés étrangères

Format :

SQL

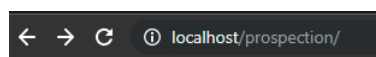
Options spécifiques au format :

Mode de compatibilité SQL : NONE

☒ Ne pas utiliser AUTO_INCREMENT pour la valeur zéro

Exécuter

Etape 8 : Il va falloir importer notre fichier *data_prospection.sql* situé dans le dossier *DataBase* du dossier *prospection*. Pour importer une base de données à partir d'un fichier externe, il faut se placer dans la nouvelle base de données créée précédemment. Ensuite, il faut sélectionner la rubrique « Importer », puis appuyer sur le bouton « Choisir un fichier », et sélectionner notre fichier *data_prospection.sql*. Après avoir appuyé sur le bouton « Exécuter », notre base de données est prête à l'utilisation.



Etape 9 : Pour lancer notre programme, il suffira d'exécuter l'adresse *localhost/prospection*. La page *home.php* s'affiche par défaut. Pour se connecter, vous pouvez tester le compte *frantzchar* mot de passe *frantzchar* (étudiant à Pau), *vohoangmai* mot de passe *vohoangmai* (compte administrateur), et *moreladrie* mot de passe *moreladrie* (étudiant à Cergy).

Attention à ne pas oublier de lancer Wamp avant d'effectuer ces neuf étapes. Après avoir fait ces étapes, veuillez redémarrer votre Wamp pour ne pas avoir de problèmes avec l'actualisation du fichier *php.ini*.

III Tâches et obligations

III.1 Répartition des tâches

L'organisation et la répartition des tâches au sein de notre groupe est un enjeu clé pour la réussite du projet service prospection proposé par ATILLA. Répartir les tâches en fonction des compétences informatiques de chacun constitue la base du travail en équipe. En tant que débutants au niveau du développement web, Adnane EL KASMI et Aboubacar KOUROUMA se sont exclusivement occupés de la partie front-END qui contient le HTML5 pour afficher le contenu à l'écran, le CSS3 pour embellir les pages HTML, le JavaScript pour gérer les actions utilisateurs, et aussi Photoshop pour créer le logo du site. Ayant également contribué au front-END, les initiés Charles-Philippe FRANTZ et Christelle VO se sont chargés de la partie back-END qui regroupe le PHP7 pour manipuler tout type de variables et produire une page dynamique via un serveur, le SQL pour gérer les bases de données, et aussi LDAP afin de pouvoir se connecter au site grâce aux mêmes identifiants que ceux utilisés sur AREL (mais les mots de passe sont ici les identifiants eux-mêmes pour des raisons de sécurité donc, pour accéder au compte utilisateur vohoangmai par exemple, le mot de passe est « vohoangmai »).

Afin que cela soit plus limpide, voici un tableau récapitulatif de la répartition du travail.

Elève / Tâche	HTML	CSS	PHP	JavaScript	SQL	LDAP	Photoshop	Rapport
Adnane EL KASMI	X	X	X	X	X		X	X
Charles-Philippe FRANTZ	X	X	X		X	X		X
Aboubacar KOUROUMA	X	X	X	X				X
Christelle VO	X	X	X	X	X			X

III.2 Exigences

- Être motivé et savoir motiver.
- Chercher à s'impliquer, quel que soit son niveau.
- Aimer organiser.
- Avoir une connaissance globale du cahier des charges.
- Avoir des compétences en HTML5, CSS3, JavaScript, PHP7, SQL.
- Savoir donner et recevoir du feedback, voire des critiques.

IV Cahier des charges

Dans le cahier des charges écrit par le service prospection, les réalisations exigées sont l'affichage en temps réel des événements disponibles pour les Eistiens, l'inscription en ligne, et l'envoi de mails automatiques. Nous pouvons alors relever quatre points importants de notre site.

Mise en page : Pour un site épuré et agréable à parcourir, nous avons privilégié les couleurs basiques (le blanc et le noir). La couleur bleue rappelle le logo de l'EISTI. La sidenav réalisée dans *index.php* est présente sur le côté gauche de toutes les pages sauf *home.php* et *login.php*.

Compte administrateur : Dans la rubrique Etablissements, l'administrateur visualise sous forme de tableaux les prospections (sous-rubrique Prospections), les établissements ayant organisé des prospections incluant des Eistiens mais qui n'en ont pas encore prévu dans l'avenir (sous-rubrique En attente), et les établissements n'ayant pas encore organisé de prospections incluant des Eistiens (sous-rubrique Pas de forum). Le cahier des charges demande que la sous-rubrique En attente affiche les établissements dont nous ne savons pas s'ils organiseront une prospection incluant des Eistiens, et que la sous-rubrique Pas de forum affiche les établissements qui ont déclaré ne pas organiser de prospections incluant des Eistiens, mais nous avons choisi de disposer de la façon dite précédemment. Dans cette rubrique Etablissements, l'administrateur peut choisir un campus ou/et filtrer ou/et ordonner pour obtenir le tableau qu'il souhaite. Dans les sous-rubriques Prospections et En attente, l'administrateur peut cliquer sur une ligne du tableau pour accéder à la fiche de l'établissement associé afin de consulter les informations de la prospection, des inscrits, des anciens, etc., et les quatre commentaires. Dans la sous-rubrique Prospections, le cahier des charges demande que la couleur rouge soit associée aux prospections ayant des problèmes, mais nous avons choisi de l'associer aux prospections ayant deux places libres. Dans la rubrique Gestion, la sous-rubrique Ajouter comporte un formulaire demandant les informations de la prospection à ajouter dans la base de données, et la sous-rubrique Supprimer comporte un formulaire demandant des informations de la prospection à supprimer de la base de données. Dans la rubrique Extraction, la sous-rubrique Exporter permet d'exporter le tableau souhaité (contactlycee, forum, ou lycee) en fichier csv, et la sous-rubrique Importer permet d'importer le fichier csv dans la base de données. Dans la rubrique Documents, l'administrateur peut ajouter, supprimer, consulter, télécharger des documents.

Compte étudiant : Contrairement à l'administrateur, l'étudiant visualise uniquement les informations associées à son campus, peut seulement consulter le premier commentaire, peut s'inscrire à une prospection, et ne peut ni ajouter ni supprimer de documents. Dans la rubrique Mes prospections, il visualise ses prospections à venir (sous-rubrique A venir), ses prospections dont le compte-rendu est vide (sous-rubrique Compte-rendu), et ses prospections dont le compte-rendu n'est pas vide (sous-rubrique Terminées).

Inscription et mail automatique : L'étudiant peut s'inscrire à condition que la prospection possède une place libre, que l'étudiant ne s'y est pas déjà inscrit, et que la prospection n'est pas déjà passée. S'il clique sur le bouton « S'inscrire », il reçoit un mail automatique confirmant son inscription et contenant les informations de la prospection choisie.

V Organisation

Dans la mesure où la moitié des membres du groupe n'avait pas de bases solides dans le développement web, la première difficulté a été de gérer la répartition des tâches. De ce fait, nous avons décidé de laisser du temps aux débutants d'acquérir un certain nombre de connaissances qui leur permettraient d'apporter leurs pierres à l'édifice.

Ainsi, après quelques temps de formation, le projet a pu débiter et la répartition des tâches a ainsi pu être décidée : le *HTML* et le *CSS* seraient gérés principalement par les anciens CPGE, le *PHP* et le *SQL* par les expérimentés du groupe. Le projet allait être ficelé comme suit :

- Un fichier *index.php* qui contient les éléments communs à toutes les pages comme la sidebar de navigation principale et le pied de page. C'est aussi lui qui appelle les pages auxquelles l'utilisateur souhaite accéder, à l'aide d'un **Controller**.
- Un fichier *HomeController.php* qui associe le nom d'un fichier php à son chemin depuis le fichier *index.php* afin de faciliter l'appel des pages.

Il semble important de signaler à ce niveau que chacune des pages du site correspond à un fichier php, rédigé par un mixte de langage php et HTML, lui-même contenu dans un dossier *Views*. L'ensemble du rendu visuel du site est aussi géré par un système de dossiers et de sous-dossiers dont les rôles sont les suivants :

V.1 Controllers

HomeController.php : Le *HomeController.php*, comme expliqué ci-dessus, facilite les redirections vers les fichiers php. La classe **HomeController** contient les méthodes à appeler en fonction du fichier php demandé.

V.2 Index.php

Le fichier *index.php* rassemble ce qui est commun aux autres fichiers php, donc elle contient notamment la sidebar ainsi que le pied de page permettant de revenir en haut de la page.

V.3 Home.php

Le fichier *home.php* est une page ajoutée (non présente dans le cahier des charges) pour la présentation du projet et des membres du groupe, ainsi que pour la consultation du cahier des charges, la redirection vers le site de l'EISTI, et l'accès à la page de connexion.

V.4 DataBase

Le dossier *DataBase* contient le fichier *data_prospection.sql* qui comporte la base de données exportée et utilisée dans notre projet. Voici des extraits de nos quatre tables :

id	Lycee	Username	Prenom	Nom	Email	Privileges	Image	Campus
1	12	frantzchar	Charles-Philippe	Frantz	frantzchar@eisti.eu	Student	Pictures/imagesTest/calopsittes.jpg	Pau
2	0	vohoangmai	Christelle	Vo	vohoangmai@eisti.eu	Admin	Pictures/imagesTest/arableu.jpg	Cergy
3	6	test1	testPrénom	testNom	testmail@eisti.eu	Student	Pictures/imagesTest/oiseaux.jpg	Cergy

FIGURE 2 – Structure et exemple de la table utilisateurs

Id	Type	Nom	Code_Postal	Departement	Ville	Proche_de	Standard	Fax	Mail_Lycee	AdressePostale	Commentaire1	Commentaire2	Commentaire3	Commentaire4
1	Lycée	Camille Claudel	95490	95	Vauréal	Cergy	testStandard1	01.23.65.45.89.75	camilleclaudel@gmail.com	4 Avenue Federico Garcia Lorca	sdjsjdi	1ceci est commentaire 2éééá	1ceci est commentaire 3éééá	1ceci est commentaire 4éééá
2	Lycée	CPP	95600	95	L'eau est bonne	Cergy	testStandard2	fax2	cppeaubonne@hotmail.com	18 RUE DU TONNELIERRRRRRR	2ceci est commentaire 1éééá	2ceci est commentaire 2éééá	2ceci est commentaire 3éééá	2ceci est commentaire 4éééá
3	IUT	NomIUT	87590	87	Villeville	Proche de nulle part	testStandard3	02.01.98.77.23	iutdevilleville@gmail.com	20 RUE DU TONNELIER	3ceci est commentaire 1éééá	3ceci est commentaire 2éééá	3ceci est commentaire 3éééá	3ceci est commentaire 4éééá

FIGURE 3 – Structure et exemple de la table lycee

Id	Lycee	DateForum	Heure_Debut	Heure_Fin	Courrier	Public	Annee_du_dernier_forum	Type_de_prospection	Participant1	Participant2	CompteRendu
1	4	2020/02/23	16h	23h	testcourrier@forum.com	tout public	2019	Forum	4	3	NULL
2	2	2019/01/10	09h	10h	courriertest1@gmail.com	ceci n'est pas tout public	2017	Conférence	4	3	test2compterendu
3	1	2017/09/01	10h	22h	courriertest2	ceci n'est pas tout public	2015	Forum	3	4	CSKODSOKDKSOKDSO

FIGURE 4 – Structure et exemple de la table forum

Id	Genre	Nom	Prenom	Fonction	Mail	MailForum	LigneDirecte	Idlycee
1	M	FRANTZ	Charles-Philippe	Secrétaire	test@test.fr	forumtest@test.fr	01.34.05.80.90	2
2	F	VO	Christelle	Directrice	test2@test.fr	forumtest2@test.fr	01.35.43.23.88	5
3	F	COLLAS	Collas	CPE	colcollas@moissan.fr	collas@moissan.fr	01.74.25.15.87	6
4	M	FRANTZ	Charles-Philippe	Secrétaire	test@test.fr	forumtest@test.fr	01.34.05.80.90	1

FIGURE 5 – Structure et exemple de la table contactlycee

Clés primaires : les clés primaires de notre base de données sont les Id de chaque table.

Clés étrangères : les clés étrangères de notre base de données sont : Idlycee dans la table contactlycee ; participant1 et participant2 dans la table forum ; Lycee dans la table utilisateurs.

V.5 JS

Le dossier *JS* contient les fichiers *JavaScript* comportant des fonctions permettant d'agir au sein d'un fichier php selon les actions de l'utilisateur.

V.6 CSS

Le dossier *CSS* contient les fichiers *CSS* permettant d'embellir le site.

V.7 Html2Pdf

Le dossier *Html2Pdf* permet de convertir un contenu html en un pdf consultable et téléchargeable. Nous aurions voulu l'utiliser pour le compte-rendu, car Charles-Philippe FRANTZ et Christelle VO ont déjà su l'utiliser pour le projet du semestre dernier afin de consulter ou/et télécharger les factures des commandes des clients mais, par manque de temps, nous avons préféré nous focaliser sur les fonctionnalités proposées dans le cahier des charges.

V.8 Uploads

Le dossier *uploads* contient les documents à consulter ou/et à télécharger mis en ligne par l'administrateur sur le site.

V.9 Cahier_des_charges

Le dossier *cahier_des_charges* contient le *cahier_des_charges.pdf*.

V.10 Les images

Les dossiers *images* (d'Aboubacar KOUROUMA), *img* (d'Adnane EL KASMI), et *Pictures* (du groupe entier) contiennent les images présentes sur le site : le logo de l'Eisti, l'icône de déconnexion, etc. Nous avons prévu de placer toutes les images dans le dossier *Pictures* mais, par manque de temps, nous avons laissé les dossiers tels quels.

V.11 Views

Le dossier *Views* comporte tous les fichiers php sauf *HomeController.php*, *index.php*, et *home.php*. Il contient :

V.11.1 Administrateurs

Le dossier *Administrateurs* comporte l'ensemble du site accessible à l'administrateur.

Documents : Ce dossier contient les fichiers suivants :

- **documents.php** : Ce fichier est une page qui permet à l'administrateur de gérer les documents en ligne. Il permet d'ajouter ou de supprimer un fichier, directement consultable ou uniquement téléchargeable par les utilisateurs du site.

- **upload.php** : Ce fichier n'est pas une page du site ; elle ne s'affiche pas à l'écran, mais permet d'ajouter un document consultable ou/et téléchargeable sur le site.
- **deletefile.php** : Similaire au fichier *upload.php*, mais s'occupant de la suppression de documents déjà présents sur le site.

Etablissements : Ce dossier contient les fichiers suivants :

- **accueil_admin.php** : Comme son nom l'indique, ce fichier correspond à la page affichée pour l'administrateur après avoir passé la page de connexion. Il affiche les prospections sous forme de tableau. L'administrateur peut choisir un campus ou/et filtrer ou/et ordonner pour obtenir le tableau qu'il souhaite. Les lignes des prospections n'ayant aucune place libre sont en jaune, les lignes de celles ayant deux places libres sont en rouge.
- **attente.php** : L'affichage sur cette page est aussi sous forme de tableau. Il contient les établissements ayant organisé des prospections incluant des Eistiens mais n'en ayant pas encore prévu dans l'avenir. L'administrateur peut choisir un campus ou/et filtrer ou/et ordonner pour obtenir le tableau qu'il souhaite.
- **noForum.php** : Ce fichier permet d'afficher la liste des établissements n'ayant pas encore organisé de prospections incluant des Eistiens. L'administrateur peut choisir un campus ou/et filtrer ou/et ordonner pour obtenir le tableau qu'il souhaite.

Fiche : Nous voulons pouvoir cliquer sur une ligne du tableau de *accueil_admin.php* ou de *attente.php* afin d'afficher plus de détails sur l'établissement et la prospection associée, comme les contacts, les inscrits, les anciens, etc. Pour cela, nous avons :

- **fiche_lycees.php** : Ce fichier permet d'afficher la page d'informations d'un établissement comme expliqué précédemment. Il va chercher ces informations dans notre base de données.
- **InfoInscritCergy.php** : Ce fichier permet d'envoyer un mail automatique à un étudiant du campus de Cergy qui vient de s'inscrire à une prospection.
- **InfoInscritPau.php** : Il s'agit du même fichier que le précédent, mais adapté au campus de Pau.
- **autres_mails_auto** : Ce dossier contient les autres mails automatiques que nous n'avons pas encore intégrés dans les fonctionnalités du site.

Gestion : Ce dossier contient les pages **ajouter.php** et **supprimer.php** qui affichent des formulaires afin d'ajouter ou de supprimer une prospection dans la base de données.

Extraction : Ce dossier contient :

- **analyse.php** : Permettant à l'administrateur d'importer un fichier csv dont les informations s'ajoutent alors à la base de données actuelle. **importCSV_SQLCONTACTLYCEE.php**, **importCSV_SQLFORUM.php** et **importCSV_SQLLYCEE.php** sont les fichiers qui aident *analyse.php* dans sa tâche.
- **extraction.php** : Permettant d'effectuer le processus inverse, c'est-à-dire exporter une table de la base de données actuelle en fichier csv.
- Les dossiers *contactlycee*, *forum*, et *lycee* contiennent chacun un fichier csv prêt à importer, ce qui permet de tester l'importation du fichier *analyse.php*.

V.11.2 Elèves

Le dossier *Elèves* comporte l'ensemble du site accessible à l'étudiant.

Documents : Le dossier *Documents* de la partie Elèves ne contient qu'un fichier php : le *documents_eleves.php* qui possède les mêmes caractéristiques que celui de l'administrateur, sauf qu'il ne permet pas d'apporter de modification aux documents présents sur le site. En effet, seul l'administrateur peut en ajouter ou en supprimer.

Établissements : Il contient les mêmes fichiers php que le dossier correspondant pour l'administrateur, sauf que l'élève peut s'inscrire selon les conditions cités dans la section Cahier des charges, qu'il a uniquement accès aux informations associées à son campus, et qu'il ne peut lire que le premier commentaire.

Forums : Ce dossier permet à l'étudiant de suivre l'évolution des prospections auxquelles il a pris part ou auxquelles il souhaiterait prendre part. Ainsi, on y retrouve les fichiers suivants :

- **mes_prospections.php** : qui affiche ses prospections à venir.
- **compte_renduphp** : qui affiche ses prospections passées dont le compte-rendu est vide.
- **terminees.php** : qui affiche ses prospections passées dont le compte-rendu n'est pas vide.

V.11.3 Autres

Login et Logout : Le dossier *Login* comporte le fichier *login.php* qui contient le formulaire de connexion (nom d'utilisateur et mot de passe) avec authentification LDAP. La partie php de ce fichier vérifie si les deux champs sont bien remplis, si le mot de passe possède au moins huit caractères, si le nom d'utilisateur et le mot de passe sont validés via l'authentification LDAP, si l'utilisateur est administrateur ou étudiant. Si l'utilisateur est reconnu dans l'authentification LDAP mais n'est pas présent dans la table utilisateurs de notre base de données, alors sa ligne correspondante sera ajoutée automatiquement. Le dossier *Logout* comporte le fichier *logout.php* qui demande à l'utilisateur s'il souhaite vraiment se déconnecter. En cas de confirmation, la session se ferme, et l'utilisateur est renvoyé à la page de connexion. Sinon, l'utilisateur revient à la page précédente.

CompteRendu : Le dossier *CompteRendu* comporte le fichier *compteRendu.php* qui affiche un tableau récapitulant le compte-rendu de la prospection. Par manque de temps, toutes les cases dédiées aux données contiennent par défaut « variable ».

VI Ce que nous avons réussi

Dans cette partie, nous allons énumérer les différentes fonctionnalités que nous avons réussi à implémenter dans notre site. La liste est la suivante :

- Authentification LDAP se trouvant dans le fichier *login.php* dans le dossier **Views** puis **Login**
- Bouton déconnexion
- Page d'accueil du site permettant de présenter le projet, son but et les membres du projet, ainsi que de pouvoir consulter le cahier des charges, aller sur le site de l'EISTI et un bouton « Se Connecter »
- Utilisateurs : compte élève et compte administrateur
- Importation de fichiers pour la base de données (fichier .csv vers base SQL)
- Exportation de tables de la base de données vers des fichiers .csv
- Uploader, télécharger, consulter et supprimer des fichiers stockés sur le site
- Sous forme de tableaux, affichage des prospections, des établissements ayant organisé des prospections incluant des Eistiens mais n'en ayant pas encore prévu dans l'avenir, des établissements n'ayant pas encore organisé de prospections incluant des Eistiens
- Pour l'étudiant, sous forme de tableaux, affichage de ses prospections à venir, de ses prospections passées dont le compte-rendu est vide, de ses prospections passées dont le compte-rendu n'est pas vide
- Différents choix de filtres et d'ordres pour l'affichage des tableaux
- Affichage de l'établissement correspondant à la ligne choisie du tableau
- Envoi automatique d'un mail à un étudiant qui s'inscrit à une prospection selon son campus
- Mise à jour du nombre de places d'une prospection lorsque des étudiants s'inscrivent
- Impossible pour un étudiant de s'inscrire à une prospection s'il n'y a plus de places disponibles, s'il s'y est déjà inscrit, ou si la prospection est déjà passée
- Un étudiant peut uniquement voir et s'inscrire aux forums de son campus associé
- Un étudiant peut uniquement voir les commentaires étudiants
- Affichage des informations des inscrits et des anciens (prénom, nom, mail, photo)
- Affichage du compte-rendu
- Formulaire pour ajouter ou supprimer une prospection en particulier

VII Problèmes rencontrés

Dans cette partie, nous allons voir les différents problèmes que nous avons rencontrés. Nous allons distinguer les problèmes résolus des problèmes en cours.

VII.1 Problèmes résolus

Formulaire (inscription et commentaire) : Nous voulions faire un formulaire qui actualise la même page php et qui, normalement, entre dans un if qui fait une action, sauf que cela ne fonctionne pas et, vu que nous n'avons pas trouvé pourquoi, nous avons fait autrement : nous avons créé une autre page php et, à la place du formulaire, nous avons fait un bouton qui renvoie vers cette nouvelle page qui active directement l'action au lieu de vérifier un if.

Choix du campus dans la sidenav : Nous n'avons pas réussi à activer le choix du campus de la sidenav dans la page courante, donc nous avons mis le menu déroulant directement dans la page courante.

Bouton retour : Au début, nous pensions stocker la page précédente dans le lien mais, en y réfléchissant, nous avons compris que, si nous appuyons une deuxième fois d'affilée sur le bouton retour, nous revenons alors à la page suivante. Depuis, nous avons trouvé une fonction JavaScript qui permet de revenir à la page précédente sans problème.

VII.2 Problèmes en cours

Formulaire (commentaire) : Comme nous avons expliqué dans la sous-section précédente, l'action du formulaire ne fonctionne pas, mais la solution d'une autre page php n'est pas possible, car il faudrait stocker le commentaire dans le lien. Or, le commentaire contient des espaces, ce qui pose problème.

Envoi de mails dépendants d'une durée : Nous avons créé les différents types de mails pour les deux campus, mais nous ne savons pas comment les envoyer à un moment précis. En effet, nous savons faire des actions uniquement lorsqu'un utilisateur se déplace dans le site donc, si nous voulons envoyer un mail à un jour fixé mais aucun utilisateur ne se connecte ce jour-là, le mail prévu ne sera pas envoyé. De plus, le contraire peut arriver : si nous voulons envoyer un mail à une seconde près mais deux utilisateurs se déplacent à ce moment-là, deux mails identiques seront envoyés.

Supprimer un document : Pour une raison que nous n'avons pas trouvée, lorsqu'il reste un seul fichier qui ne fait pas partie du dossier **Both**, l'icône poubelle renvoie vers la page de connexion au lieu de permettre la suppression du document associé.

VIII Axes d'améliorations

Arrivé à la date butoir, voici la liste de quelques points que nous aurions voulu finaliser.

Tout d'abord, nous aurions aimé terminer les fonctionnalités pratiques du site qui sont notamment :

- **Les mails** : Bien que nous ayons réussi l'envoi automatique d'un mail à un étudiant qui s'inscrit à une prospection, il faudrait le faire également pour informer l'établissement associé, ou pour d'autres actions comme l'annulation, le rappel, etc.
- **La rubrique Etablissements** : Les établissements en attente doivent pouvoir passer dans la liste *pas de forum* ou celle *forum disponible* (prospections) après validation ou invalidation par l'administrateur.
- **fiche.lycees.php** : Les informations en haut à droite de la fiche ne sont pas encore gérées. Les propositions de modification de la fiche en haut à droite et de commentaires ne sont pas fonctionnelles. Le nombre de personnes l'an passé et le compte-rendu ne sont pas encore liés à la base de données. L'année de la prospection précédente ne se met pas à jour lorsqu'une prospection du même établissement est ajoutée.
- **La rubrique Gestion** : Les formulaires d'ajout et de suppression ne sont pas encore fonctionnels.
- **La rubrique Mes prospections** : L'étudiant ne peut pas encore remplir le compte-rendu en ligne.

Ensuite, nous aurions voulu pouvoir améliorer le rendu visuel du site. Nous avons en stock trois différents rendus possibles du site, mais nous n'avons pas eu le temps de trier ou de fusionner avec le rendu final. En effet, nous nous sommes surtout concentrés sur l'incorporation d'un maximum de fonctionnalités, en donnant au site une apparence qui permet surtout de cerner son fonctionnement. De même, nous avons prévu de placer toutes les images dans le dossier *Pictures* mais, par manque de temps, nous avons laissé les dossiers *images* et *img* tels quels. De plus, nous aurions voulu donner la possibilité de consulter ou/et télécharger le compte-rendu en pdf via le dossier *Html2Pdf*, car Charles-Philippe FRANTZ et Christelle VO ont appris à le faire lors du projet du semestre dernier pour consulter ou/et télécharger les factures des commandes des clients.

Et finalement, sur le plan sécurité :

- **Le backpage** : Il faudrait pouvoir empêcher le retour à la session fermée sans avoir à entrer les identifiants et le mot de passe, en appuyant seulement sur le bouton retour.
- **Les zones de texte** : Il s'agit de s'assurer que, si un utilisateur entre un quelconque langage informatique à des fins inattendues, le site puisse le remarquer et l'en empêcher.

IX Conclusion

Pour conclure, en plus de répondre aux réalisations exigées (affichage des prospections, inscription, mail automatique), notre site possède plusieurs fonctionnalités proposées dans le cahier des charges (fiche établissement) et d'autres de notre initiative (rubrique gestion). Par conséquent, nous avons rempli notre but principal qui est de proposer au service prospection une plateforme en ligne qui est plus pratique et lisible que le système de partage de documents actuel, qui permet aux étudiants de s'inscrire directement en ligne et de recevoir un mail automatique confirmant cette inscription, qui met à jour le nombre de places, et qui distingue les prospections passées de celles à venir.

Ainsi, nous sommes satisfaits du site web que nous avons créé. Nous avons anticipé de nombreux problèmes, ce qui a permis de trouver rapidement les conditions d'inscription, de savoir d'avance (sans avoir testé) que le fonctionnement du bouton retour devait être changé, etc. Malgré cela, nous pensons que, avec plus de temps, nous pourrions l'améliorer ainsi qu'ajouter des fonctionnalités. Ce projet est particulièrement motivant car les clients sont réels et le cahier des charges est détaillé, ce qui nous fait sentir utiles et efficaces. De plus, ce projet a été pour nous un bon aperçu de ce que serait un travail de groupe en entreprise (savoir faire participer chaque membre, quel que soit son niveau, et intégrer les travaux de chacun au rendu final), et a permis de nous améliorer rapidement, ce qui donne l'impression qu'il reste encore beaucoup à apprendre en développement web.