

NOLLE Damien
1^{ère} Année BTS SIO (Option : SLAM)
Etablissement Saint-Adjutor (Vernon)
Promotion : 2021 – 2022

RAPPORT DE STAGE



Ensemble Scolaire Catholique
— Sainte Agnès - Jeanne d'Arc —

Entreprise d'accueil : Ensemble Scolaire Catholique de Sainte Agnès – Jeanne d'Arc

Période de stage : 30 mai au 1^{er} juillet 2022

Tuteur/rice de stage : Madame HELLARD

Enseignant référent : Madame DALLIER

Enseignant de visite : Monsieur GAULIER

Remerciements :

Je tiens à remercier l'équipe pédagogique de l'Ensemble Scolaire Catholique de Sainte Agnès – Jeanne d'Arc à Vernon de m'avoir accueilli et fait confiance pour la remise à niveau de leur site web sur la gestion des anciens élèves de l'établissement.

Je tiens aussi à remercier Madame HELLARD d'avoir été ma tutrice de stage et d'avoir eu le temps et la patience de m'accompagner tout le long du stage.

Je tiens aussi naturellement à remercier toute l'équipe pédagogique du BTS SIO de Saint-Adjutor à Vernon de m'avoir accueilli pour cette première année et de m'avoir apporté les compétences et les connaissances nécessaires à la réalisation de ce stage.

Présentation de l'entreprise :

L'Ensemble Scolaire Catholique (ESC) de Sainte Agnès – Jeanne d'Arc est un établissement scolaire privé réunissant école maternelle et primaire, collège, lycée et études supérieures. Dans le cadre du stage, nous allons nous concentrer sur le lycée et les études supérieures. Deux baccalauréats professionnels sont proposés : Métiers du Commerce et de la Vente (MCV) et Gestion et Administration (GA). Ce dernier ne sera plus proposé l'année prochaine, il ne restera donc que le Bac Pro MCV. Une 3^{ème} prépa-métiers est proposée au collège et, en terme d'études supérieures, un CAP Équipier Polyvalent du Commerce (EPC) et deux BTS, Management Commercial Opérationnel (MCO) et Négociation et Digitalisation de la relation Client (NDRC) sont proposés. Cet établissement propose donc principalement des formations orientées professionnelles et sur le commerce, le management et la vente.

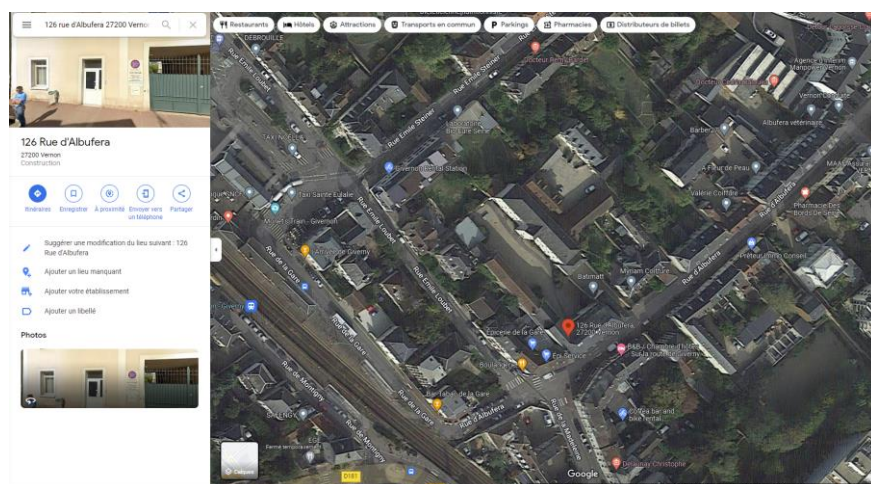
D'un point de vue ressource humaine, une équipe pédagogique de 22 professeurs pour 160 élèves au lycée. Au vu de la fusion des Organismes de Gestion des Etablissement Catholiques (OGEC) de Vernon (Comprenant dans un premier temps, en 2011, le collège Jeanne d'Arc et les établissements de Sainte Agnès, une autre fusion est prévue pour l'année prochaine avec l'établissement Saint-Adjutor), l'équipe de gestion se retrouve fusionnée avec Monsieur THOMASSIN en tant que directeur, Madame HELOIN en tant qu'adjointe du directeur, Madame HELLARD en tant que coordinatrice de l'Unité de Formation et d'Apprentissage (UFA en études supérieures), Madame PONT en secrétaire, Madame BOUTTARD en CPE et Monsieur DOUZI en tant que professeur documentaliste du CDI.

En termes de matériels et de bâtiments, l'établissement dispose d'un bureau d'accueil où la secrétaire et Madame HELLARD disposent de leurs bureaux, de trois bâtiments disposant de salles de cours (les bâtiments sont nommés par une lettre (C, A, B) et disposent chacun de 10 salles de cours ou autre) et de matériels informatiques. En termes de logiciels pour les étudiants, on parle principalement de suite pour de la bureautique (Microsoft 365) au vu de la nature de leurs formations.

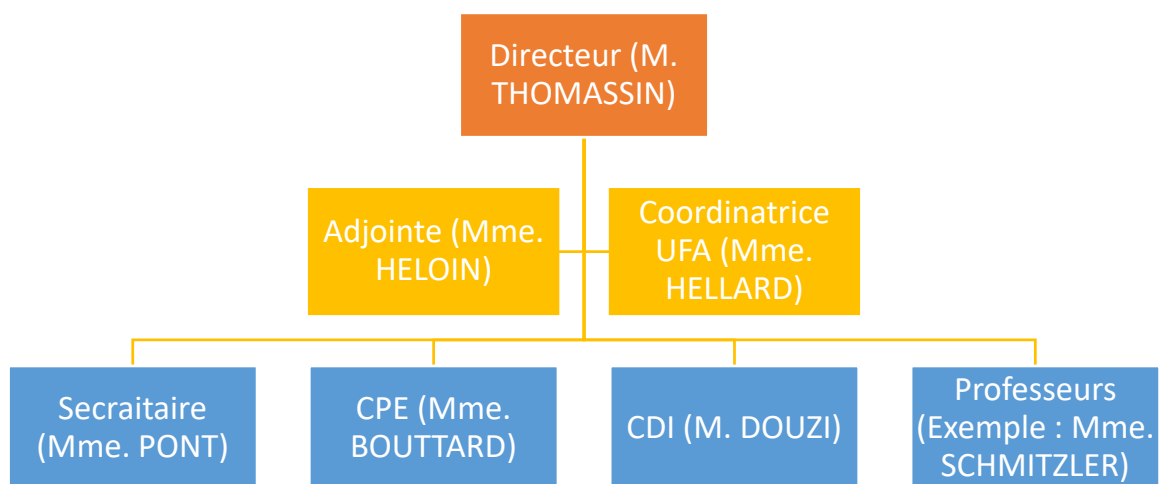


Le directeur, Monsieur THOMASSIN, devant l'ESC Sainte Agnès – Jeanne d'Arc.

L'établissement se situe en plein centre de Vernon à l'adresse 126 rue d'Albufera 27200 :

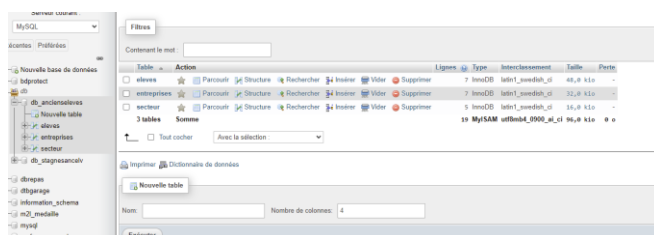


Organigramme de l'ESC Sainte Agnès Jeanne d'Arc :



Présentation du service d'accueil et des moyens informatiques :

Mon stage s'est principalement réalisé en distanciel, les moyens mis en place sont donc mes propres moyens à disposition dont mon propre poste informatique. Un ordinateur portable contenant deux existants (une base de données et une première version du site) m'a été prêté afin que je puisse les récupérer.



Voici les deux existants récupérés, la base de données et la première version du site.

En termes de logiciels, j'avais accès à WAMP Server qui est un serveur Web fonctionnant sous Windows et qui permet d'héberger des sites en local. Il est composé de tous les outils nécessaires pour la création et l'hébergement de sites Web (PHPMyAdmin, Apache, ...). J'avais aussi accès à Visual Studio Code, un logiciel de programmation (uniquement en code) gratuit développé par Microsoft qui est idéal pour la programmation Web notamment grâce à sa simplicité d'utilisation.



Logo de WAMP Server.



Logo de Visual Studio Code.

Pour pouvoir entrer en contact avec ma tutrice de stage afin d'obtenir des conseils, de l'aide ou pour la tenir au courant de l'avancement du projet, le logiciel Teams par Microsoft a été utilisé afin d'établir nos communications



Logo de Microsoft Teams.

Des rendez-vous à l'ESC Sainte Agnès – Jeanne d'Arc et à l'établissement Saint-Adjutor ont été réalisés afin de suivre le projet ou pour obtenir de l'aide ou des informations. L'équipe pédagogique et administrative de l'ESC Sainte Agnès – Jeanne d'Arc sont restées à ma disposition en cas de besoin.

Les langages utilisés pour le développement web sont : PHP, HTML, CSS, JavaScript et SQL.



Présentation du sujet :

Madame SHMIZLER, professeur principale de CAP et enseignante au lycée, décide de réaliser avec l'une de ses classes un site Web archive et gestionnaire des anciens élèves de l'établissement. Le site web devait permettre à l'enseignant et aux élèves de saisir des données sur ces anciens élèves tels que leurs noms, prénoms, moyens de communications, anciennes filières et emplois actuels afin de les archiver et de permettre de les consulter. Cette application Web devait aussi permettre de pouvoir envoyer des messages à ces anciens élèves au cas où ils étaient sollicités pour des prestations en tant qu'anciens élèves au sein de l'établissement par exemple où encore permettre la recherche d'informations et de données à l'aide de filtres. Pour l'accès à l'application Web, un système d'authentification par couple identifiant/mot de passe et par profil (administrateur, professeur et élève) devait être mis en place.

Un étudiant en deuxième année du BTS SIO à Saint-Adjutor était chargé de s'occuper de ce projet mais au vu des nombreux soucis rencontrés par Madame SHMIZLER pour utiliser l'application Web, il m'est demandé de reprendre ce projet.

Trois objectifs principaux ont été mis en place, à savoir : refaire la base de données, refaire l'application Web et m'occuper de l'hébergement. Le cahier des charges restant le même que pour cet étudiant en deuxième année à quelques changements près.

Cahiers des charges :

- Refaire l'application Web en méthode MVC.
- Chercher un style graphique représentant l'établissement et le réaliser en Web Responsive.
- Faire un système d'authentification par couple identifiant mot de passe et par profil (administrateur, professeur et élève).
- Faire un système de gestion des anciens élèves de l'établissement et des utilisateurs de l'application Web (Ajout, modifier et supprimer) en interagissant avec la base de données.
- Faire un système de filtrage permettant de naviguer à travers les données avec des critères de tri.
- Faire un système de messagerie permettant de contacter par mail ces anciens élèves.
- S'occuper de l'hébergement de l'application Web.

SOMMAIRE :

I) Base de données :

1. Étude de l'existant (reprise en retro-conception).
2. Critique de la base de données.
3. Modification.

II) Application Web :

1. Démarrage et création du projet.
 - a. Création de la structure MVC.
 - b. Découvertes et installation des librairies importantes pour ce projet.
2. Partie Authentification.
3. Partie Gestionnaire.
4. Partie Messagerie.

III) Hébergement

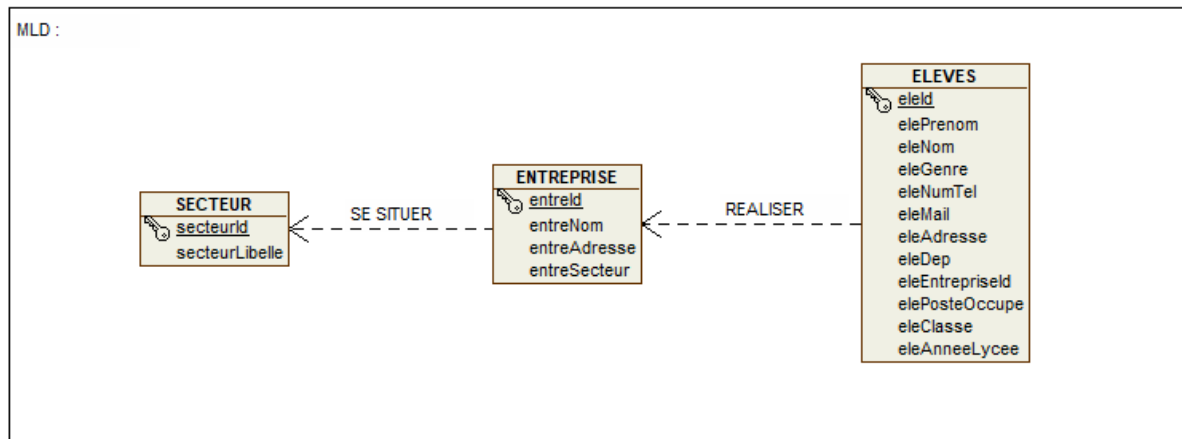
IV) Conclusion

V) Annexes

Clé Étrangère : eleEntrepriseld en référence à entreld(Entreprise)

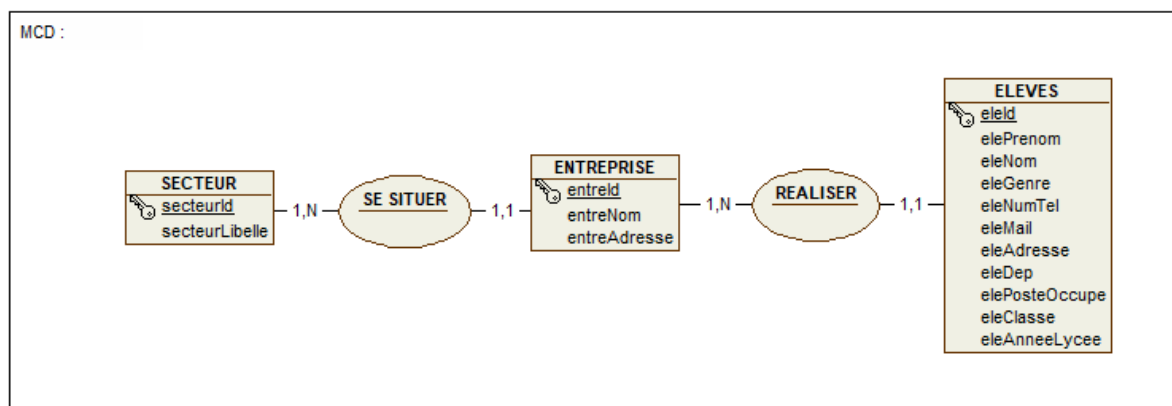
Maintenant que les différentes tables et relations ont été mise en évidence avec le MRD ou le SRD, il faut réaliser le MLD (Modèle Logique de Données) qui reprend les éléments du SRD ou du MRD mais de façon plus graphique, ce qui permettra par la suite de réaliser le MCD (Modèle Conceptuel de Données) et de finir la retro-conception de la base.

MLD :



Une fois le MLD terminé, il suffit de le reprendre mais en MCD. Il permet de mettre en évidence à la fois les relations entre les tables de la base de données grâce à des associations et aux clés primaires et étrangères tout en étant graphique.

MCD :



Grâce à cette étape de rétro-conception, je vais pouvoir faire la critique de cette base de données et venir apporter des améliorations et des modifications.

I) Base de données

2. Critique de la base de données

On remarque grâce à la rétro conception et grâce à l'existant, que la colonne « eleClasse » de la table « eleves », contenant les filières, contient des doublons tout comme pour la colonne « elePosteOccupe », contenant les postes actuels occupés par les anciens élèves dans les entreprises. Il faudrait créer une entité pour ces deux colonnes comme pour le secteur contenant ainsi un libellé et un identifiant.

Il n'y a pas d'entité permettant de faire le lien entre l'entreprise, les élèves et le poste, il faudrait venir créer une entité qui contiendrait le lien entre ces trois table en contenant leurs identifiants en clé primaire et étrangère.

Les colonnes « eleDep » et « eleAdresse » de la table « eleves » contiennent toutes les deux le code postal. Il faudrait supprimer la colonne « eleDep » et diviser « eleAdresse » en trois colonnes (rue, ville et code postal) afin d'éviter ce doublon tout en plus que ce découpage permet de renforcer la sécurité de la base de données. Il faudrait faire ce même découpage pour la colonne « eleAdresse » de la table « entreprises ».

Aucune table n'est présente pour les utilisateurs, il faudrait donc prévoir une table à cette effet pour pouvoir contenir l'identifiant, le mot de passe (crypté) et le type de droit. (1 pour les administrateurs, 2 pour les professeurs et 3 pour les élèves). Le type de droit sera contenu dans une entité à part comme le poste, le secteur et la filière.

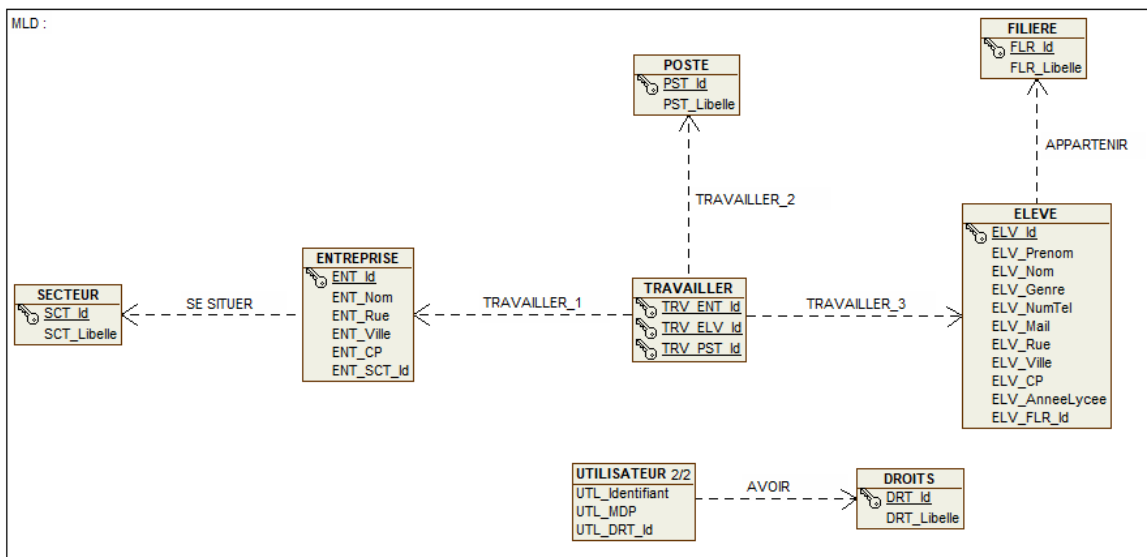
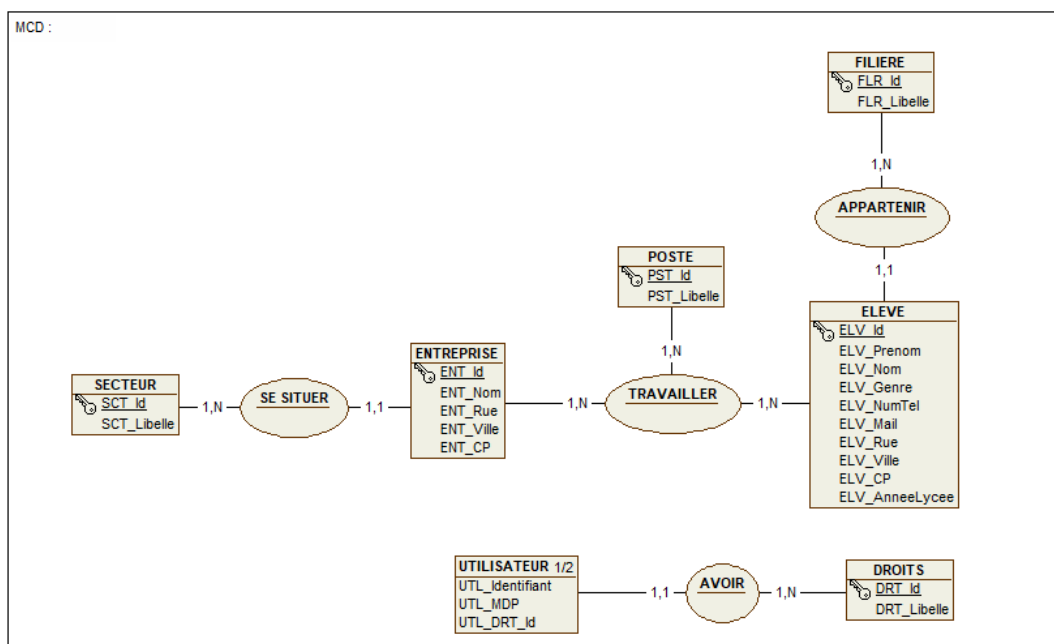
Il n'y a pas de nomenclature standardisée, pour les noms des tables, il faudrait commencer par « tb_ ». Pour les noms des colonnes et des contraintes (clé primaire, étrangère), il faut commencer par les 3 lettres du nom de la table ou de la contrainte (exemple : « tb_eleves » → « elv_ » ou encore « tb_entreprise » → « ent_ ») et le nom de la colonne. Les clés primaires sont nommée « PK + [Numéro clé] » (Exemple : « PK1 ») et les clés étrangères « FK + [Numéro clé] » (Exemple : « FK1 »).

I) Base de données

3. Modification

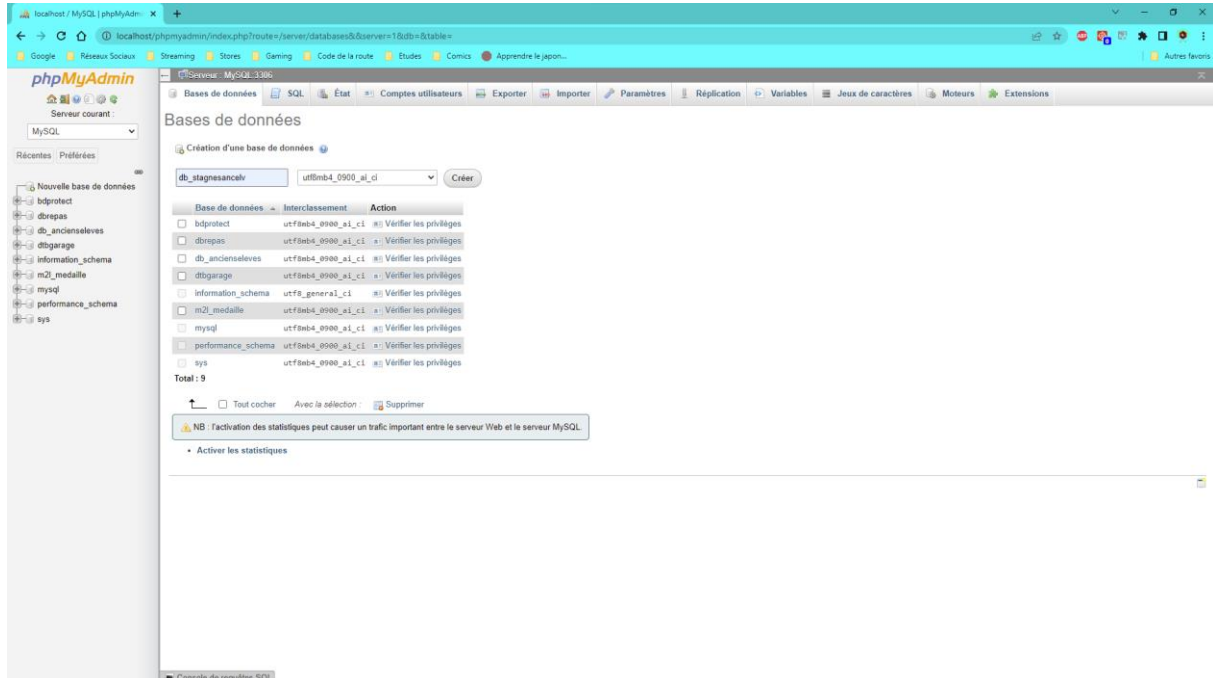
Grâce à la critique de la base de données réalisée précédemment, je vais pouvoir réaliser le MLD et le MCD de cette nouvelle base afin de pouvoir, par la suite, construire la base de données.

MCD et MLD de la base de données modifiés :



Une fois ce MCD et MLD validés par ma tutrice de stage, j'ai pu construire la nouvelle base de données en Requêtes SQL sur PHPPMyAdmin.

Dans un premier temps, la base de données a été créée :



Puis les différentes tables ont été créées avec leurs clés primaires, grâce à des requêtes SQL :

- **Création de la table pour les secteurs :**

```
CREATE TABLE tb_secteur
```

```
(
```

```
    SCT_Id int NOT NULL AUTO_INCREMENT,
```

```
    SCT_Libelle varchar(30) NOT NULL,
```

```
    CONSTRAINT PK1 PRIMARY KEY (SCT_Id)
```

```
)
```

Modification du Stockage Engine vers InnoDB pour pouvoir créer les clés étrangères plus facilement par la suite car sinon cela peut créer des problèmes lors de la création de ces clés et changement de l'interclassement vers latin1_swedish_ci pour l'import et la compatibilité des anciennes données sur cette nouvelle base de données :

Options pour cette table

Renommer la table en ☒ [Ajuster les privilèges](#)

Commentaires de la table

Moteur de stockage InnoDB

Interclassement latin1_swedish_ci ☒ [Changer le classement de toutes les colonnes](#)

AUTO_INCREMENT

ROW_FORMAT COMPACT

- **Création de la table pour les entreprises :**

CREATE TABLE tb_entreprise

(

ENT_Id int NOT NULL AUTO_INCREMENT,

ENT_Nom varchar(30) NOT NULL,

ENT_Rue varchar(200) NOT NULL,

ENT_Ville varchar(50) NOT NULL,

ENT_CP int NOT NULL,

ENT_SCT_Id int NOT NULL,

CONSTRAINT PK2 PRIMARY KEY (ENT_Id)

)

Modification du Stockage Engine vers InnoDB et changement de l'interclassement vers latin1_swedish_ci :

Options pour cette table

Renommer la table en ☒ [Ajuster les privilèges](#)

Commentaires de la table

Moteur de stockage InnoDB

Interclassement latin1_swedish_ci ☒ [Changer le classement de toutes les colonnes](#)

AUTO_INCREMENT

ROW_FORMAT COMPACT

- **Création de la table pour les postes :**

```
CREATE TABLE tb_poste  
(  
    PST_Id int NOT NULL AUTO_INCREMENT,  
    PST_Libelle varchar(40) NOT NULL,  
  
    CONSTRAINT PK3 PRIMARY KEY (PST_Id)  
)
```

Modification du Stockage Engine vers InnoDB et changement de l'interclassement vers latin1_swedish_ci :

Options pour cette table

Renommer la table en ☒ Ajuster les privilèges ⓘ

Commentaires de la table

Moteur de stockage ⓘ

Interclassement ☒ Changer le classement de toutes les colonnes

AUTO_INCREMENT

ROW_FORMAT

Exécuter

- **Création de la table pour les élèves :**

```
CREATE TABLE tb_eleve  
(  
    ELV_Id int NOT NULL AUTO_INCREMENT,  
    ELV_Prenom varchar(15) NOT NULL,  
    ELV_Nom varchar(15) NOT NULL,  
    ELV_Genre char NOT NULL,  
    ELV_NumTel varchar(12) NOT NULL,  
    ELV_Mail varchar(40) NOT NULL,  
    ELV_Rue varchar(200) NOT NULL,  
    ELV_Ville varchar(50) NOT NULL,  
    ELV_CP int NOT NULL,  
    ELV_AnneeLycee year NOT NULL,  
    ELV_FLR_Id int NOT NULL,  
  
    CONSTRAINT PK4 PRIMARY KEY (ELV_Id)  
)
```

Modification du Stockage Engine vers InnoDB et changement de l'interclassement vers latin1_swedish_ci :

Options pour cette table

Renommer la table en ☒ Ajuster les privilèges

Commentaires de la table

Moteur de stockage InnoDB

Interclassement latin1_swedish_ci ☒ Changer le classement de toutes les colonnes

AUTO_INCREMENT

ROW_FORMAT COMPACT

Exécuter

- **Création de la table pour les filières :**

```
CREATE TABLE tb_filiere  
(  
    FLR_Id int NOT NULL AUTO_INCREMENT,  
    FLR_Libelle varchar(20) NOT NULL,  
  
    CONSTRAINT PK5 PRIMARY KEY (FLR_Id)  
)
```

Modification du Stockage Engine vers InnoDB et changement de l'interclassement vers latin1_swedish_ci :

Options pour cette table

Renommer la table en ☒ Ajuster les privilèges [?](#)

Commentaires de la table

Moteur de stockage [?](#) InnoDB

Interclassement ☒ Changer le classement de toutes les colonnes

PACK_KEYS

☐ CHECKSUM

☐ DELAY_KEY_WRITE

AUTO_INCREMENT

ROW_FORMAT

Exécuter

- **Création de la table pour les utilisateurs :**

```
CREATE TABLE tb_utilisateur  
(  
    UTL_Identifiant varchar(255) NOT NULL,  
    UTL_MDP varchar(255) NOT NULL,  
    UTL_Droits int NOT NULL  
)
```


Modification du Stockage Engine vers InnoDB et changement de l'interclassement vers latin1_swedish_ci :

Options pour cette table

Renommer la table en ☒ Ajuster les privilèges

Commentaires de la table

Moteur de stockage InnoDB

Interclassement latin1_swedish_ci ☒ Changer le classement de toutes les colonnes

PACK_KEYS DEFAULT

☐ CHECKSUM

☐ DELAY_KEY_WRITE

AUTO_INCREMENT

ROW_FORMAT DYNAMIC

- **Création de la table « travailler » qui va faire le lien entre les entreprises, les élèves et les postes :**

CREATE TABLE tb_travailler

(

TRV_ENT_Id int NOT NULL,

TRV_ELV_Id int NOT NULL,

TRV_PST_Id int NOT NULL,

CONSTRAINT PK6 PRIMARY KEY (TRV_ENT_Id,TRV_ELV_Id,TRV_PST_Id)

)

Modification du Stockage Engine vers InnoDB et changement de l'interclassement vers latin1_swedish_ci :

Options pour cette table

Renommer la table en ☒ Ajuster les privilèges

Commentaires de la table

Moteur de stockage InnoDB

Interclassement latin1_swedish_ci ☒ Changer le classement de toutes les colonnes

PACK_KEYS DEFAULT

☐ CHECKSUM

☐ DELAY_KEY_WRITE

AUTO_INCREMENT

ROW_FORMAT FIXED

- **Création de la table pour les droits :**

```
CREATE TABLE tb_droits  
(  
    DRT_Id int NOT NULL AUTO_INCREMENT,  
    DRT_Libelle varchar(15) NOT NULL,  
  
    CONSTRAINT PK7 PRIMARY KEY (DRT_Id)  
)
```

Modification du Stockage Engine vers InnoDB et changement de l'interclassement vers latin1_swedish_ci :

Options pour cette table

Renommer la table en ☒ Ajuster les privilèges ⓘ

Commentaires de la table

Moteur de stockage ⓘ InnoDB ▼

Interclassement latin1_swedish_ci ▼ ☒ Changer le classement de toutes les colonnes

PACK_KEYS DEFAULT ▼

☐ CHECKSUM

☐ DELAY_KEY_WRITE

AUTO_INCREMENT

ROW_FORMAT DYNAMIC ▼

Exécuter

Une fois toutes les tables créées, il faut venir ajouter les contraintes de clé étrangère pour pouvoir finir la partie relationnelle de la base de données.

- **Création de la contrainte de clé étrangère pour la table des entreprises :**

```
ALTER TABLE tb_entreprise  
ADD CONSTRAINT FK1 FOREIGN KEY (ENT_SCT_Id) REFERENCES tb_secteur(SCT_Id)
```

- **Création des contraintes de clés étrangères pour la table « travailler » :**

```
ALTER TABLE tb_travailler  
ADD CONSTRAINT FK2 FOREIGN KEY (TRV_ENT_Id) REFERENCES tb_entreprise(ENT_Id);
```

```
ALTER TABLE tb_travailler
```

```
ADD CONSTRAINT FK3 FOREIGN KEY (TRV_ELV_Id) REFERENCES tb_eleve(ELV_Id);
```

```
ALTER TABLE tb_travailler
```

```
ADD CONSTRAINT FK4 FOREIGN KEY (TRV_PST_Id) REFERENCES tb_poste(PST_Id)
```

- **Création de la contrainte de clé étrangère pour la table des élèves :**

```
ALTER TABLE tb_eleve
```

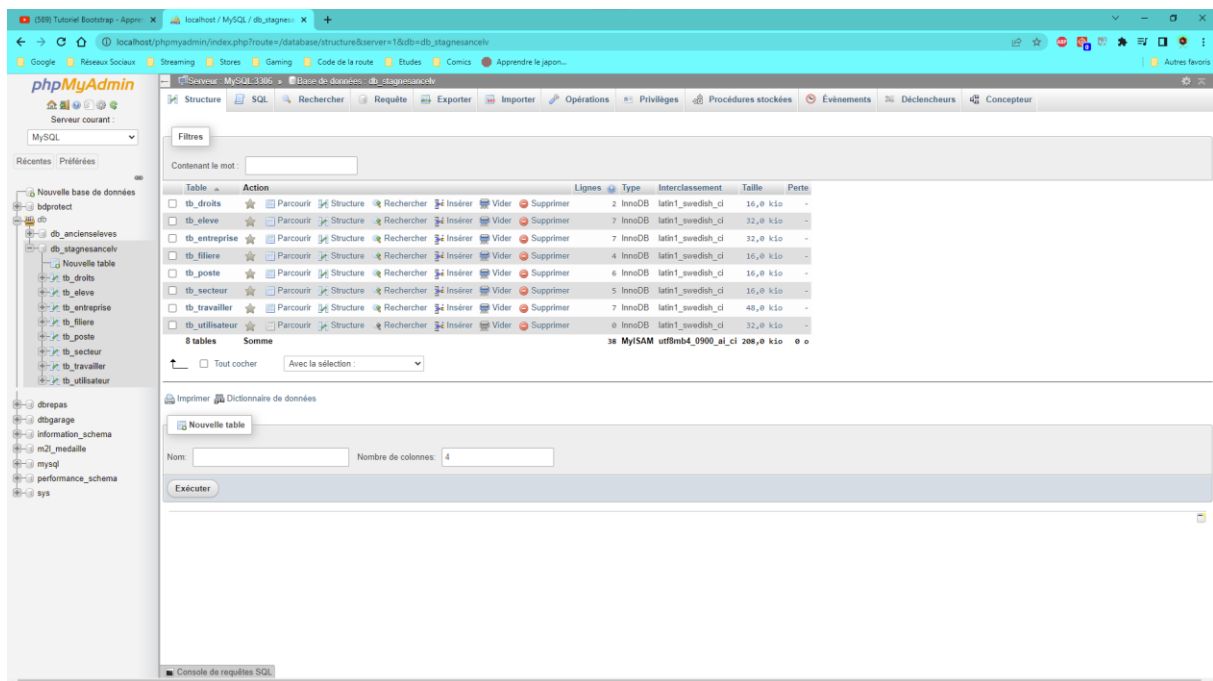
```
ADD CONSTRAINT FK5 FOREIGN KEY (ELV_FLR_Id) REFERENCES tb_filiere(FLR_Id)
```

- **Création de la contrainte de clé étrangère pour la table des utilisateurs :**

```
ALTER TABLE tb_utilisateur
```

```
ADD CONSTRAINT FK6 FOREIGN KEY (UTL_DRT_Id) REFERENCES tb_droits(DRT_Id)
```

Pour finir, les données de l'ancienne base ont été importées dans la nouvelle grâce aussi à des requêtes SQL (requêtes SQL présente dans l'Annexe 1 : « Importation des données dans les tables en requêtes SQL »). La nouvelle base de données est maintenant terminée et prête à être utilisée.



Résultat final de la base de données.

II) Application Web

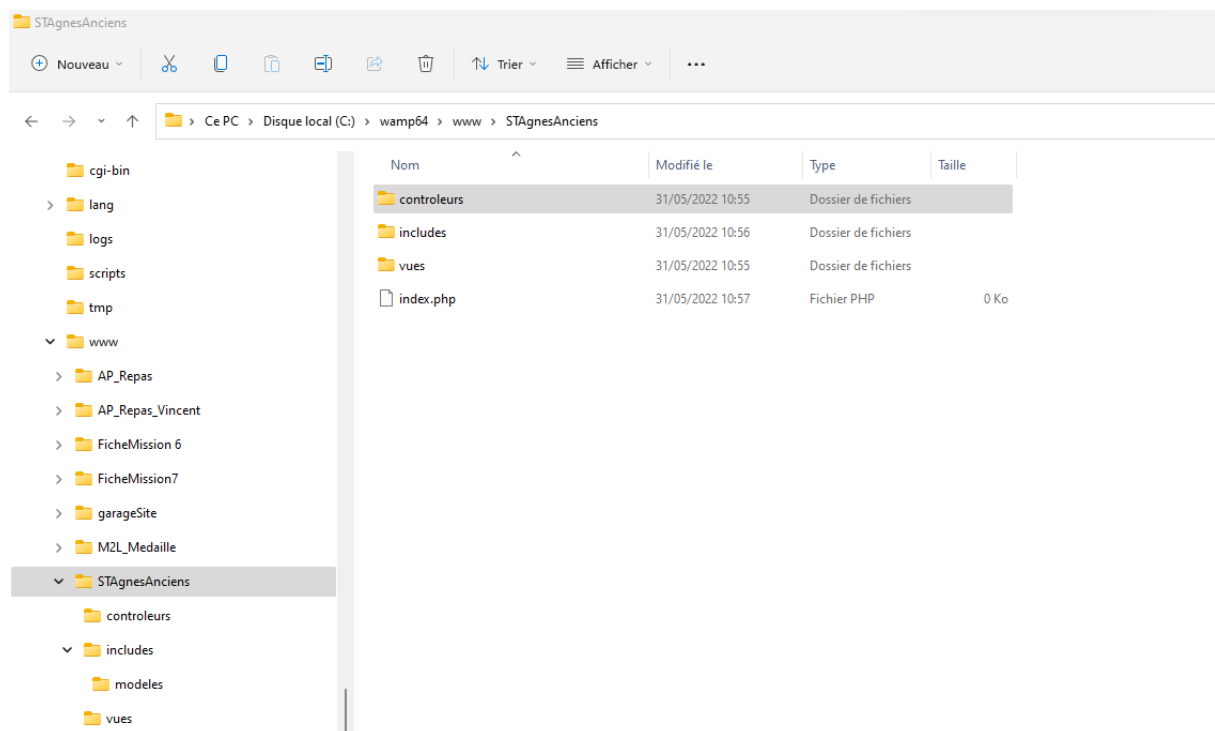
1. Démarrage et création du projet

a. Création de l'architecture MVC de l'application Web :

Une architecture MVC (Model View Controller) est une façon de décomposer une application en trois parties distinctes :

- « Model » : Partie qui va contenir la partie logique de l'application permettant principalement l'accès ou la gestion des données.
- « View » : Partie qui va contenir la partie graphique (Interfaces Homme-Machine) de l'application.
- « Controller » : Partie qui va contenir la partie dynamique de l'application. Elle permet notamment de gérer les interactions entre l'utilisateur et le reste de l'application.

Premièrement, nous allons donc créer les répertoires adéquats à cette architecture, à savoir, « controleurs », « includes » (ce répertoire contiendra un dossier nommé « models » ainsi que toutes les autres ressources (images, librairies, CSS, JS, etc...) nécessaires à l'application Web) et « view » :



Un fichier « index.php » est créé à la racine du projet qui sera donc notre point de départ de l'application Web et qui permettra de « démarrer » l'application Web et de rediriger l'utilisateur au point de départ de l'application.

Création et programmation des fichiers principaux de la méthode MVC :

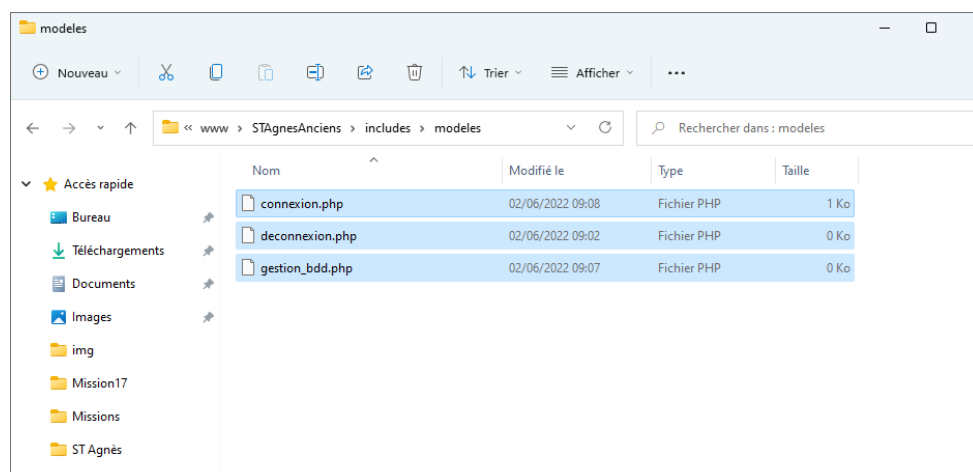
- Le contrôleur principal : « c_principal.php » est le fichier mère de toute la partie dynamique du site. Il va permettre de switch entre les différents contrôleurs afin que l'utilisateur puisse accéder aux fonctionnalités voulues. Dans ce contrôleur, la variable « \$uc » est déclarée pour la première fois. Cette variable permet de contenir le nom du contrôleur voulu permettant, grâce au « switch », d'être redirigé au contrôleur souhaité. Ayant déclaré une nouvelle session dans l'index, cette valeur sera donc toujours existante tout au long de la navigation du site. Une conditions if else est présente pour vérifier si cette variable contient quelque chose. Si oui, alors la variable garde son contenu pour que le « switch » le redirige, sinon la variable prend par défaut la variable « accueil » lui permettant d'être redirigé vers le contrôleur « c_accueil.php » :

```
1 <?php
2     if (!isset($_REQUEST['uc']))
3     {
4         $uc = "accueil" ;
5     }
6     else
7     {
8         $uc = $_REQUEST['uc'] ;
9     }
10
11     switch ($uc)
12     {
13         case 'accueil' : { include "c_accueil.php" ; break ;}
14     }
15 ?>
```

- Dans « index.php », nous allons démarrer une nouvelle session et inclure « c_principal.php » :

```
1 <?php
2     session_start();
3     include "controleurs/c_principal.php";
4 ?>
```

- Création des fichiers « connexion.php », « deconnexion.php » et « gestion_bdd.php » dans le dossier « includes/modeles » permettant l'intégration de la base de données à l'application Web :



- « connexion.php » permet d'essayer de se connecter à la base de donnée (d'où le « try ») via les paramètres de la base de données (placés dans les variables commençant par « \$PARAM_ ») et si il y a une erreur, le message et le code de l'erreur sont récupérés (d'où le « catch ») puis affichés :

```
includes > modeles > connexion.php
1  <?php
2  $PARAM_hote='localhost';
3  $PARAM_port='';
4  $PARAM_nom_bd='db_stagnancesancy';
5  $PARAM_utilisateur='root';
6  $PARAM_mot_passe='';
7
8  try
9  {
10     $bdd = new PDO('mysql:host='.$PARAM_hote.';dbname='.$PARAM_nom_bd, $PARAM_utilisateur, $PARAM_mot_passe);
11     $bdd->exec("SET CHARACTER SET utf8");
12 }
13
14 catch(Exception $e)
15 {
16     echo 'Erreur : '.$e->getMessage().'\n />';
17     echo 'N° : '.$e->getCode();
18 }
19 ?>
```

- « deconnexion.php », quant-à-lui, permet de se déconnecter de l'application « de manière générale ». Il permet d'effacer les variables contenues dans la session (« \$_SESSION ») ou bien dans la dernière variable envoyée (« \$_REQUEST »), de détruire la session, de déconnecter la base (« \$bdd=null; ») et de retourner dans l'index (« header("Location:../../index.php"); ») ce qui aura pour effet de retourner sur la page d'accueil :

```
includes > modeles > deconnexion.php
1  <?php
2  session_start();
3  $_SESSION = array();
4  $_REQUEST = array();
5  session_destroy();
6  $bdd=null;
7  header("Location:../../index.php");
8  ?>
9
```

- Inclure « connexion.php » et « gestion_bdd.php » dans « index.php » pour pouvoir les utiliser par la suite lorsque l'on souhaite faire appel à des fonctions dans « gestion_bdd.php » pour pouvoir réaliser des traitements sur les données de la base :

```
index.php
1  <?php
2  session_start();
3  include "controleurs/c_principal.php";
4  include "includes/modeles/connexion.php";
5  include "includes/modeles/gestion_bdd.php";
6  ?>
```

- Ajout dans le contrôleur principal, une « case » de la variable « \$uc » qui permettra donc cette déconnexion lorsque « \$uc » sera égal à « deconnexion » en incluant le fichier « deconnexion.php » :

```
controleurs > c_principal.php
1  <?php
2      if (!isset($_REQUEST['uc']))
3      {
4          $uc = "accueil" ;
5      }
6      else
7      {
8          $uc = $_REQUEST['uc'] ;
9      }
10
11     switch ($uc)
12     {
13         case 'accueil' : { include "c_accueil.php" ; break ;}
14         case 'authentication' : { include "c_authentication.php" ; break ;}
15         case 'deconnexion' : { include "includes/modeles/deconnexion.php"; break;}
16     }
17 ?>
```

II) Application Web

1. Démarrage et création du projet

b. Découvertes et installation des librairies importantes pour ce projet :

Découverte de Bootstrap :

Bootstrap, c'est quoi ?

Bootstrap est un Framework permettant de créer des sites responsive (adaptation du site d'un point de vue graphique sur tous les écrans comme les smartphones, tablettes, ordinateurs, etc...) de manière simplifiée et rapide.

Comment fonctionne Bootstrap ?

De manière générale, Bootstrap fonctionne comme une extension ou, plus exactement, comme une librairie qui vient se rattacher au site sous la forme de fichiers CSS qui contiennent des éléments graphiques et de design pré-faits. Plus précisément, il fonctionne sous la forme d'un système de grille et de colonnes où l'on va venir placer des éléments graphiques qui s'adapteront en fonction de l'écran et des options choisies pour s'adapter à celui-ci. Bootstrap contient aussi des éléments graphiques pré-faits permettant de faciliter la conception graphique du site.

Comment installer Bootstrap ?

Il existe différentes façons d'installer et d'utiliser Bootstrap. La manière la plus simple et la plus sûre consiste à télécharger Bootstrap et à l'installer dans le projet puis de le joindre à celui-ci aux pages où va intervenir Bootstrap.

Comment l'utiliser ?

Tout comme l'installation, il existe différentes manières d'utiliser Bootstrap. Dans tous les cas, le concept reste de lier le projet à Bootstrap. On peut soit utiliser la version JavaScript de Bootstrap, soit la version uniquement CSS ou la version en ligne. Dans tous les cas, le fonctionnement et les fonctionnalités resteront les mêmes. Tout est une question de niveau et de confort avec les langages. Il faut aussi savoir que Bootstrap propose deux versions de son outil (que ça soit la version CSS, JavaScript ou en ligne) : Une version complète qui contient toutes les fonctionnalités que peut proposer Bootstrap et une dite « minified » qui contient uniquement le strict nécessaire pour faire du Web responsive. Tout dépend de ce qu'on a réellement besoin, du stockage disponible et de la puissance du serveur. Il y a aussi des Source Map utilisable avec des outils de développement Web spécifiques et des thèmes pré-faits.

La meilleure méthode consiste à utiliser la version CSS que l'on va lier à la page où Bootstrap va intervenir car cette version permet, en plus, de créer un autre fichier CSS qui va intervenir sur les éléments de Bootstrap utilisés et permet de les personnaliser à souhait.

Concrètement ?

Une fois Bootstrap installé dans le projet, on va venir le lier au fichier où il va intervenir avec la balise :

« <link rel="stylesheet" href="[Chemin source où se trouve Bootstrap]/[bootstrap.css ou bootstrap.min.css]"> »

Puis on va créer un autre fichier CSS pour les modifications sur les éléments de Bootstrap comme expliqué précédemment et le lier avec la même balise au même fichier que la précédente, après la dite précédente balise (**/!\ ET PAS AVANT /!**).

Qu'apporte Bootstrap ?

Bootstrap apporte essentiellement des classes pré-faites et des outils permettant la réalisation d'un site Web responsive et la création d'éléments graphiques pour le design du site. Il faut donc comprendre et apprendre qu'elles sont ces différentes classes et comment les appeler dans le code HTML. Sur le site web de Bootstrap, une documentation technique explique tout le fonctionnement de Bootstrap et des différentes classes et outils disponibles.

(lien : <https://getbootstrap.com/docs/5.2/getting-started/introduction/>)

Pour réaliser ce rapport, je me suis aidé de cette documentation technique et de la vidéo de Code & Design présentant les bases de Bootstraps.

(lien : <https://www.youtube.com/watch?v=gm2RCfjXS3s&t=1145>)

Qu'elles sont les bases de bootstrap ?

Comme dit précédemment, Bootstrap utilise des classes pré-faites et des outils pour la réalisation de design pour du Web Responsive. Ils en existent énormément mais certaines sont plus importantes que d'autres à connaître et comprendre car elles seront souvent utilisées pour poser la base de la construction graphique du site Web. On va donc en parler ici et les présenter :

- container et container-fluid :

Le container-fluid est une classe permettant de contenir différents éléments graphiques qui prendra la totalité de la taille de la fenêtre tandis que le container dispose d'une taille prédéfinie. Il existe différents types de container pour les différents types d'écran (Exemples : container-sm, container-md, etc...) mais utiliser le container par défaut permet d'être responsive et donc d'adapter le contenu graphique en fonction de la taille de la fenêtre.

Pour les utiliser, il faut les appeler grâce à une balise div :

<div class="[container ou container-fluid] [Extension du nom qui permet de le modifier indépendamment dans un autre fichier CSS]">

[Contenue]

</div>

On peut ensuite soit modifier le CSS de Boostrops, soit utiliser l'autre fichier CSS précédemment créé pour modifier les propriétés des containers (Exemple : la taille en fonction de l'écran, couleur, etc...) en l'appelant via l'extension du nom donné.

Les containers permettent donc de contenir le contenu du site.

Les classes suivantes permettent d'interagir avec la grille et de la définir afin de pouvoir mettre en forme le contenu. Elles viennent donc généralement se placer dans les containers.

- row : Permet de définir une ligne.

```
<div class="row">  
    [Contenu]  
</div>
```

- col : Permet de définir une colonne de la ligne.

```
<div class="col-[type d'écran]-[taille de la colonne] col-... [On peut mettre plusieurs col d'affilée  
afin de mieux gérer le Web responsive]">  
    [Contenu]  
</div>
```

Type d'écran :

md : Écran taille moyenne, intermédiaire et grands écrans.

lg : Grand écrans et intermédiaires.

xl : Grand écrans.

sm : Petits écrans.

(Pour plus d'information, voir la documentation au lien suivant : <https://getbootstrap.com/docs/5.2/layout/containers/>)

Attention : Il n'est pas obligatoire d'utiliser les trois classes pour faire la grille, on peut très bien faire seulement container – col par exemple.

Il faut savoir qu'il existe une classe « grid » qui permet de faire la même chose que container – row – col mais de façon plus simplifiée et plus rapide.

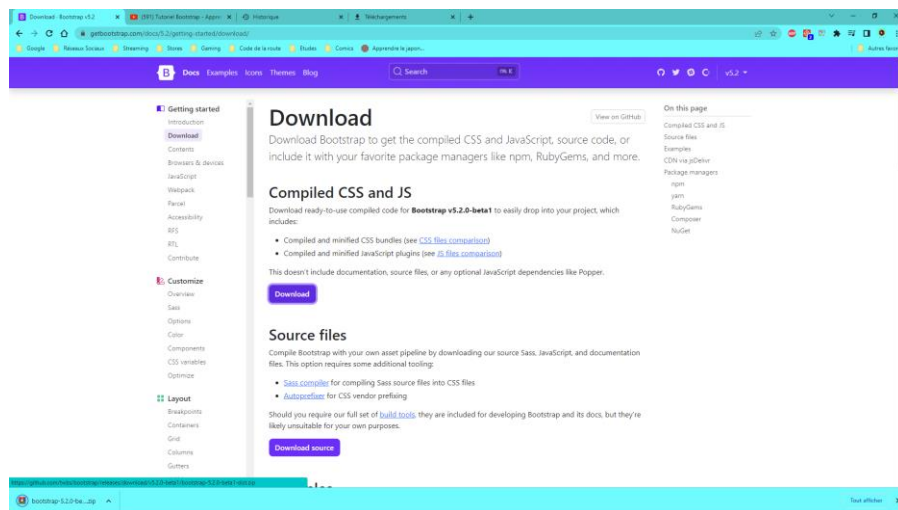
(Attention : à bien utiliser le grid en fonction de la version Bootstrap choisie, pour la version CSS, se renseigner sur le lien suivant : <https://getbootstrap.com/docs/5.2/layout/css-grid/>)

Ces classes sont les principales permettant de poser et de créer la base graphique que l'on peut aussi appeler Layout ou Grille. Il est important de se familiariser avec cette grille car c'est le point de départ de l'utilisation de Bootstrap qui va permettre de poser, comme dit précédemment, la base graphique de façon à ce que le projet soit Web responsive. D'autres classes permettant la modification ou la création de la grille sont disponibles dans la documentation de Bootstrap et dans l'onglet « Layout ». Pour découvrir Bootstrap, la meilleure méthode consiste à pratiquer tout en feuilletant la documentation ou en découvrant les fichiers CSS.

Le mieux pour découvrir Bootstrap reste de pratiquer avec la documentation et un navigateur pour des recherches en parallèle.

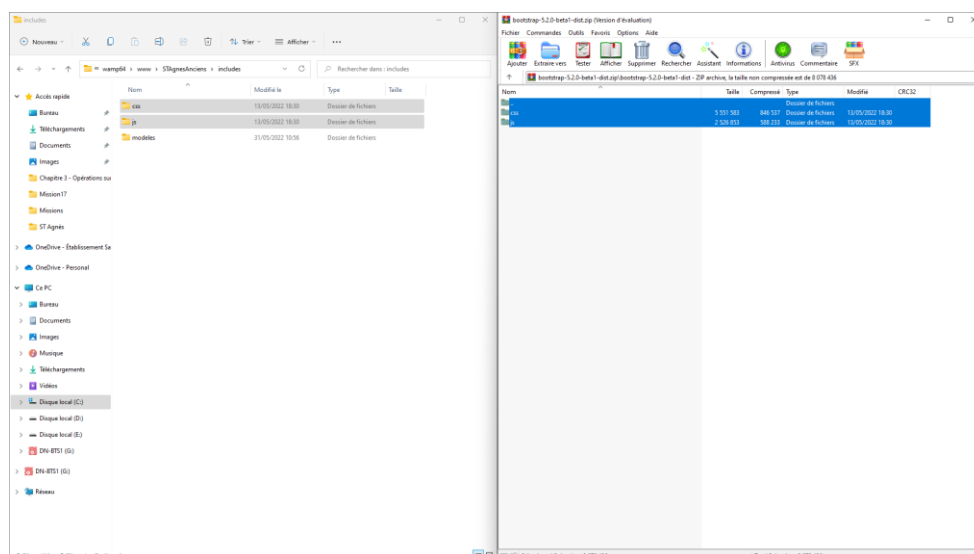
Installation de Bootstrap :

Téléchargement de Bootstrap via le site web officiel :



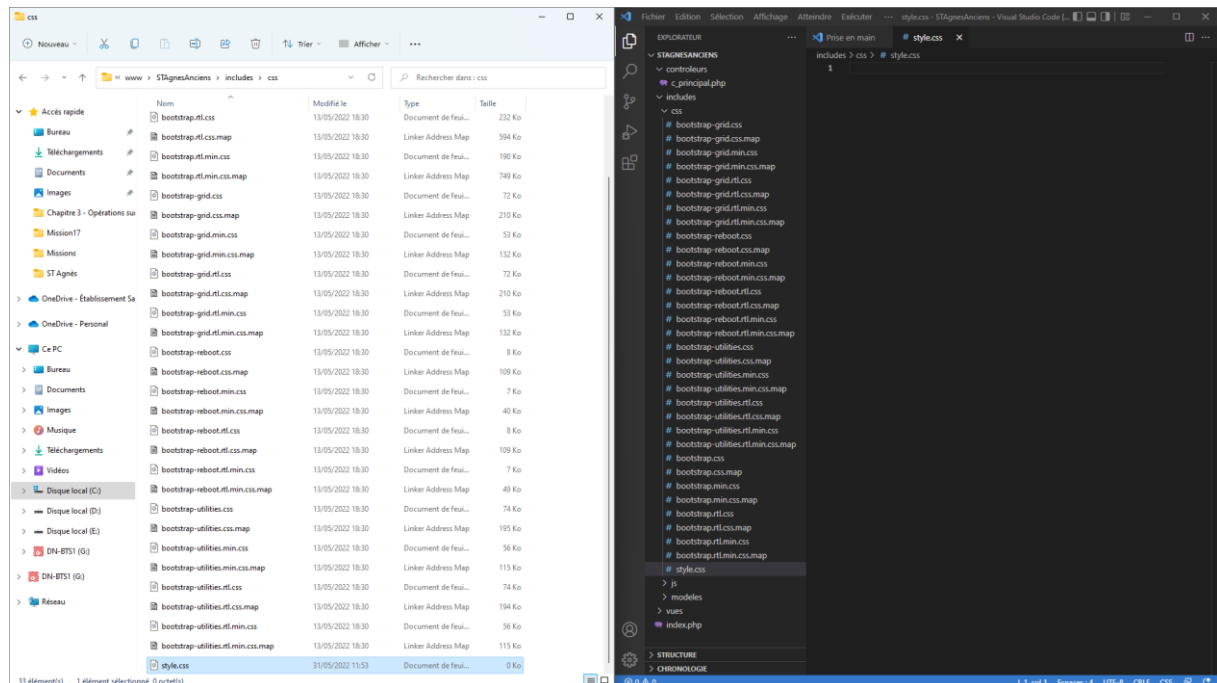
(Site web officiel : <https://getbootstrap.com/docs/5.2/getting-started/download/>)

Installation de Bootstrap dans le projet :



On extrait le contenu de l'archive ZIP précédemment téléchargée dans le dossier « includes » du projet.

Dans le dossier précédemment extrait nommé « css », on va créer un fichier « style_[Nom].css » qui permettra d'apporter des modifications aux éléments utilisés de Bootstrap :



Bootstrap est installé et prêt à être utilisé. Il ne manque plus qu'à le lier aux interfaces (« view ») où Bootstrap va venir intervenir avec la balise `<link rel="stylesheet" href="includes/css/bootstrap.css">` **EN PREMIER** et la balise `<link rel="stylesheet" href="includes/css/style-[Nom].css">` **ENSUITE**.

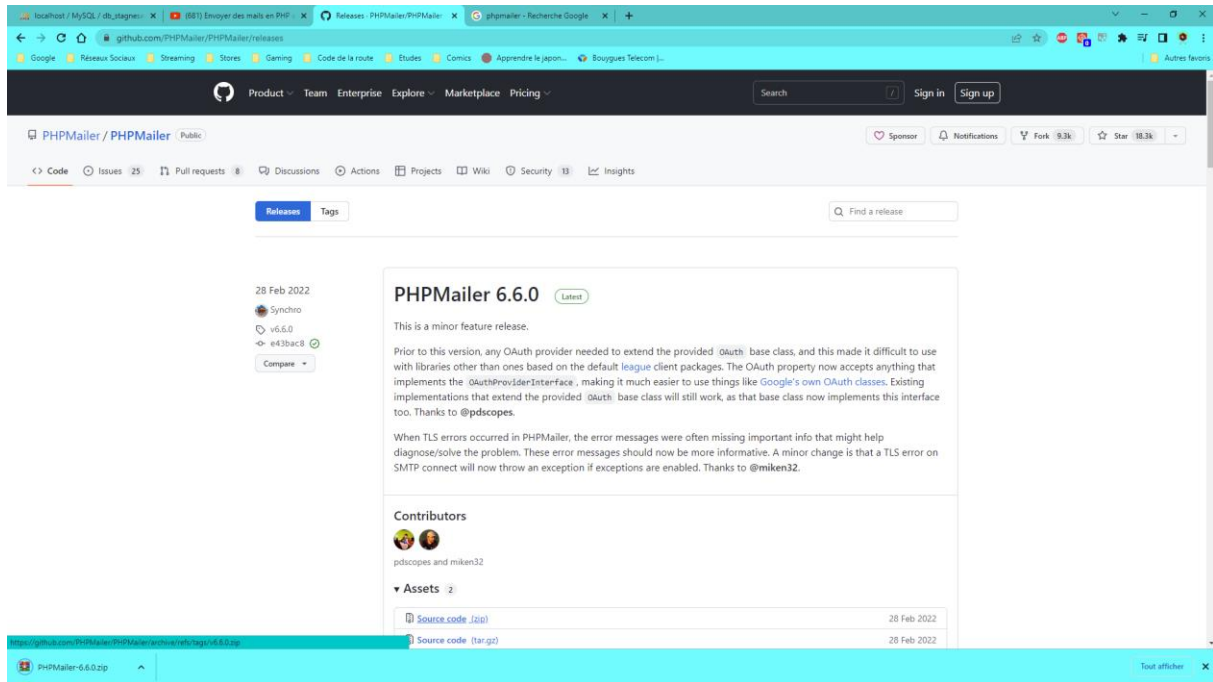
Découverte de PHPMailer :

PHPMailer, tout comme Bootstrap, est une extension ou une librairie (ou plus exactement une bibliothèque) qui vient s'installer dans le projet. L'installation et l'utilisation est donc exactement la même que Bootstrap (On télécharge PHPMailer, on l'extrait dans le projet, on l'inclue dedans, etc...) à la différence près qu'ici il fonctionne de la même manière que la bibliothèque pour la gestion des BDD.

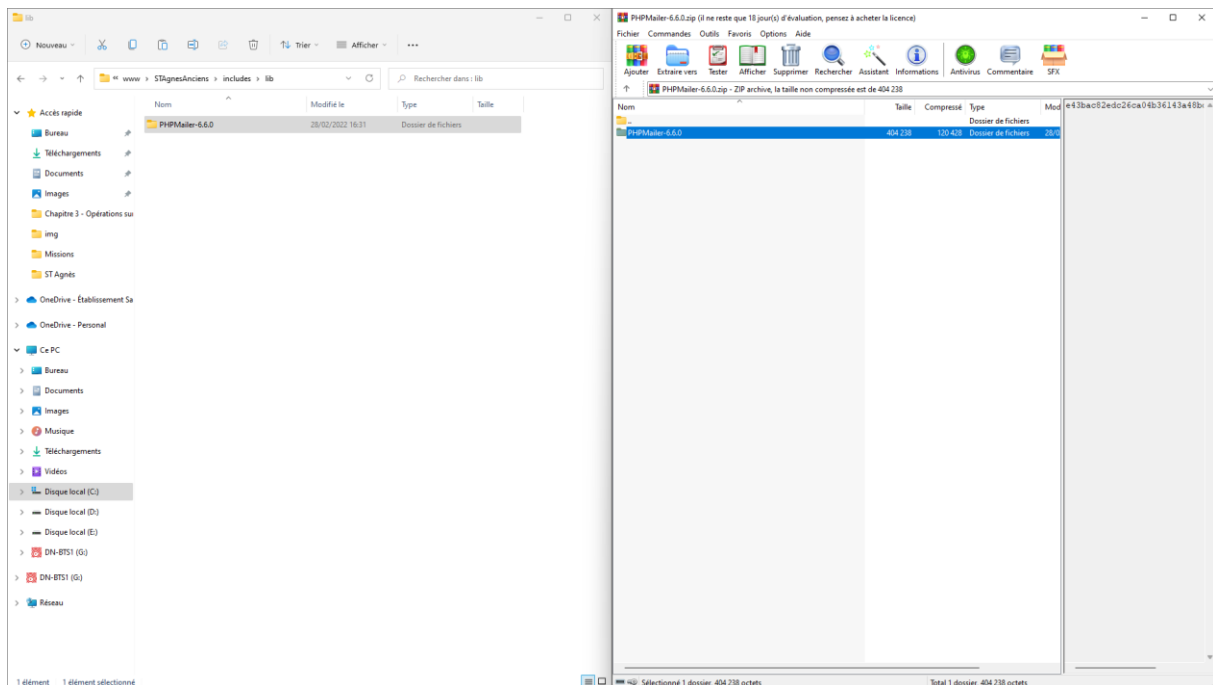
PHPMailer permet, grâce au SMTP, d'envoyer des mails en PHP. Il permet donc soit de faire des mails automatiques, soit d'intégrer une messagerie au projet.

Installation de PHPMailer :

Téléchargement de PHP Mailer via le GitHub officiel :

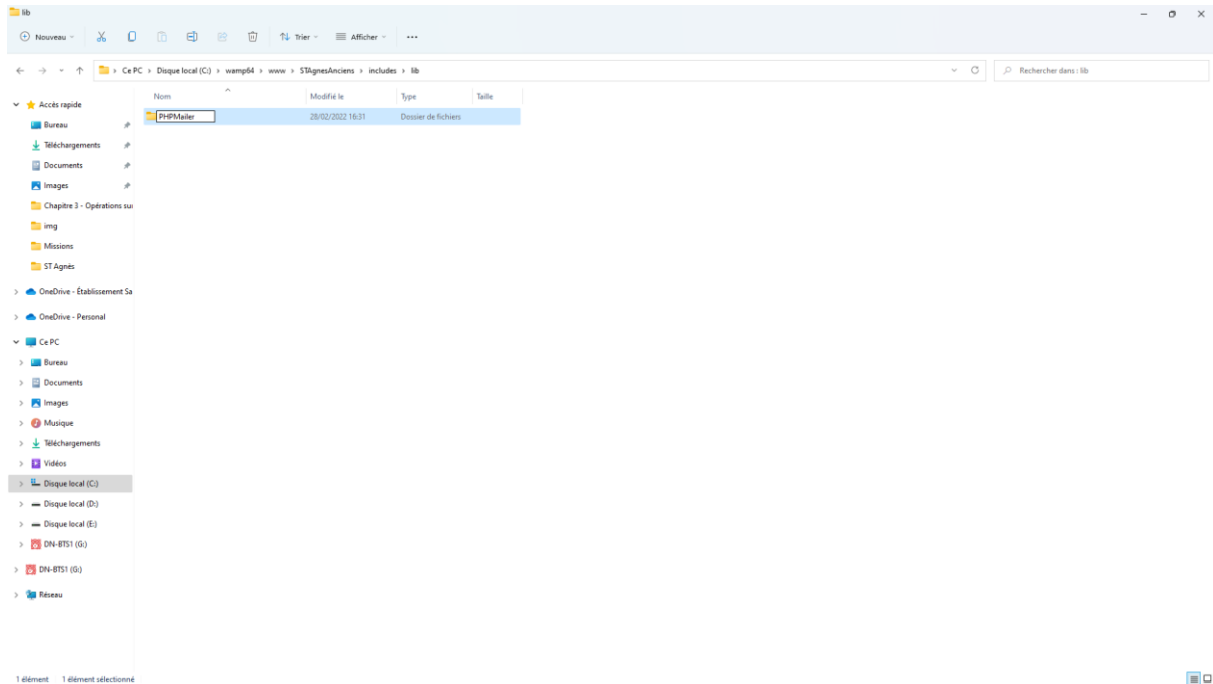


Installation de PHPMailer dans le projet :



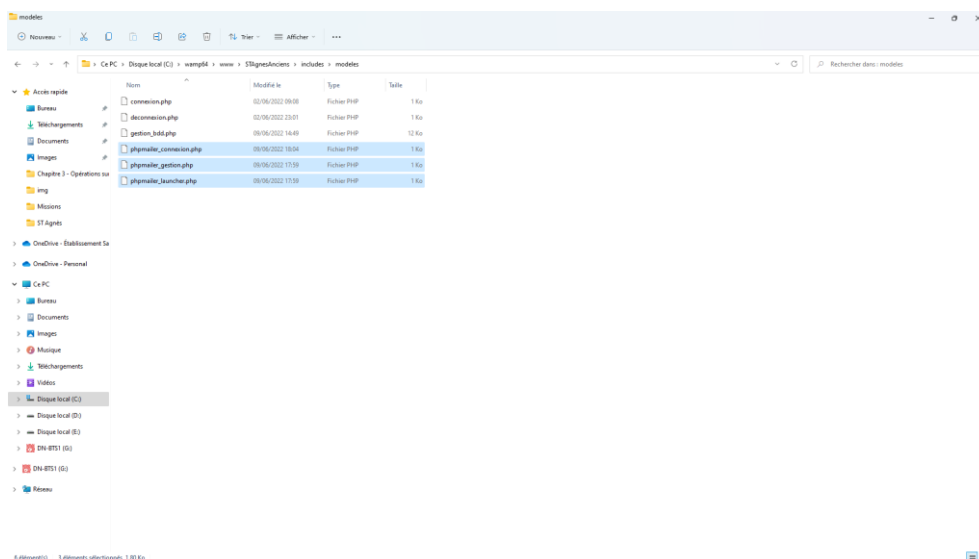
On l'installe dans un dossier « lib » dans « includes » pour pouvoir facilement le gérer de manière indépendante.

On va renommer le dossier de PHPMailer qui contenait sa version en « PHPMailer » pour pouvoir faciliter les futures installations de mise à jour :



Une fois PHPMailer d'installé, on va venir créer trois fichiers php dans « modeles » qui sont :

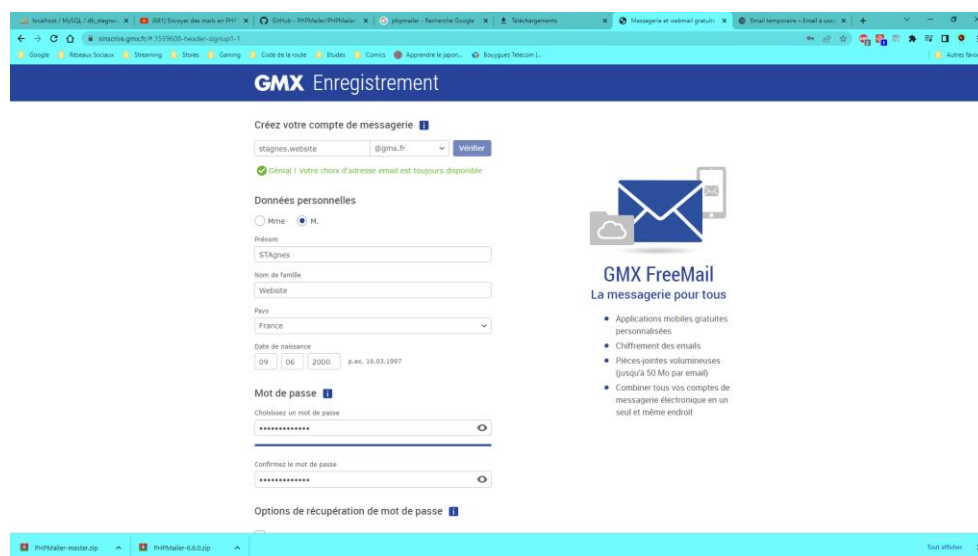
- « phpmailer_launcher.php » : il va permettre de joindre PHPMailer au projet.
- « phpmailer_connexion.php » : il va permettre de « démarrer » PHPMailer et de se connecter au serveur SMTP (à « require » dans chaque fonction de « phpmailer_gestion.php » comme pour « connexion.php » dans « gestion_bdd.php »).
- « phpmailer_gestion.php » : il va permettre de réunir les fonctions utilisant PHPMailer.



« phpmailer_launcher.php » va permettre de joindre PHPMailer au projet en « require » des fichiers avec des fonctionnalités/class qui seront utilisées dans le projet :

```
includes > modeles > phpmailer_launcher.php
1  <?php
2      require "includes/lib/PHPMailer/src/PHPMailer.php";
3      require "includes/lib/PHPMailer/src/SMTP.php";
4      require "includes/lib/PHPMailer/src/Exception.php";
5  ?>
```

On va donc venir créer une adresse E-mail qui va « servir » de serveur SMTP, dans mon cas, je vais venir créer une adresse e-mail GMX :

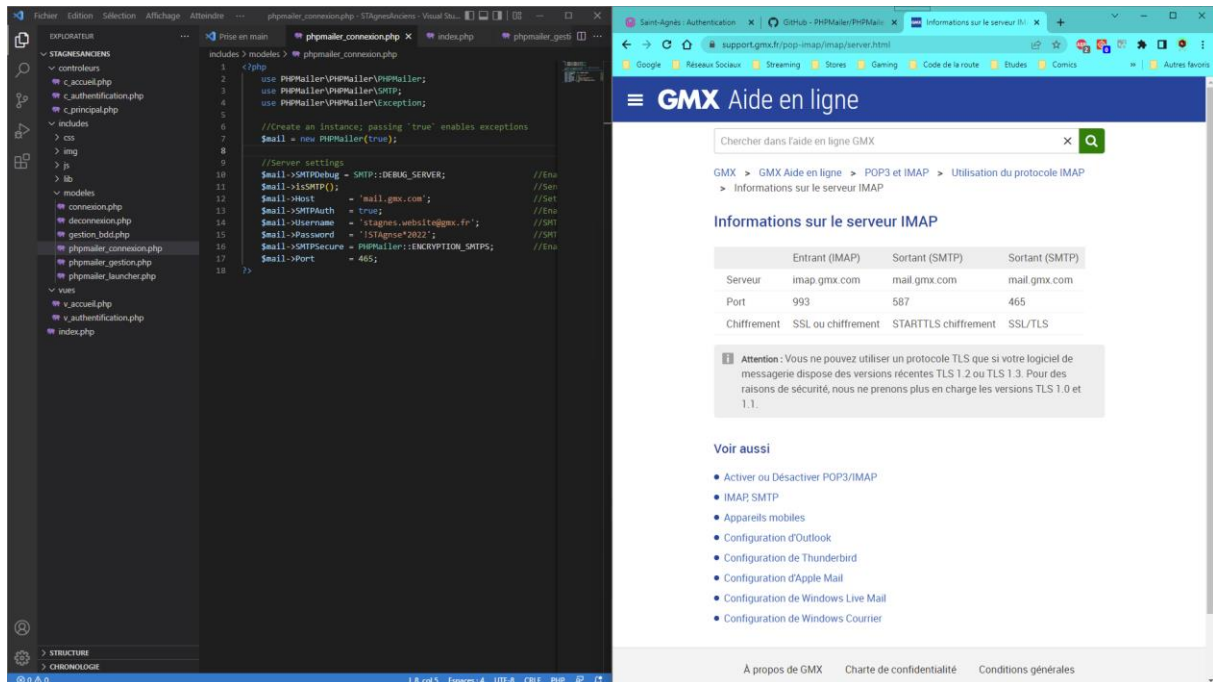


The screenshot shows the GMX registration page. The main heading is "GMX Enregistrement". Below it, there's a section "Créez votre compte de messagerie" with a text input for the email address (stagnes.website@gmx.fr) and a "Vérifier" button. A green checkmark indicates the email is available. To the right, there's a graphic for "GMX FreeMail" with the tagline "La messagerie pour tous" and a list of features: Applications mobiles gratuites personnalisées, Chiffrement des emails, Pièces jointes volumineuses (jusqu'à 50 Mo par email), and Combiner tous vos comptes de messagerie électronique en un seul et même endroit. Below the email address field, there's a section "Données personnelles" with radio buttons for "Même" or "Nouveau", and input fields for Name (STagnes), Surname, Website, Country (France), and Date of birth (09/06/2000). There's also a "Mot de passe" section with fields for password and confirmation, and a link for "Options de récupération de mot de passe".

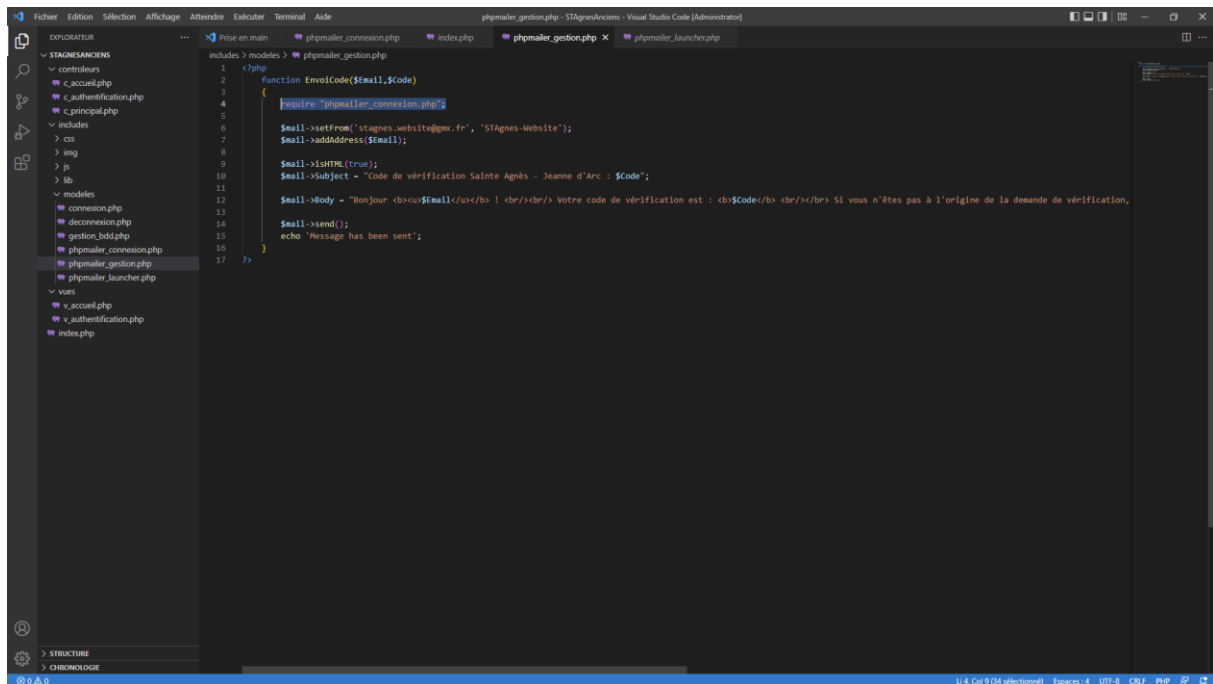
« phpmailer_connexion.php » va contenir les déclarations de class de PHPMailer et les paramètres de connexion au serveur SMTP ainsi que la déclaration de la variable « \$mail » qui est importante pour utiliser PHPMailer dans le projet :

```
includes > modeles > phpmailer_connexion.php
1  <?php
2      use PHPMailer\PHPMailer\PHPMailer;
3      use PHPMailer\PHPMailer\SMTP;
4      use PHPMailer\PHPMailer\Exception;
5
6      //Create an instance; passing `true` enables exceptions
7      $mail = new PHPMailer(true);
8
9      //Server settings
10     $mail->SMTPDebug = SMTP::DEBUG_SERVER;           //Enable verbose debug output
11     $mail->isSMTP();                                 //Send using SMTP
12     $mail->Host       = 'mail.gmx.com';               //Set the SMTP server to send through
13     $mail->SMTPAuth   = true;                         //Enable SMTP authentication
14     $mail->Username    = 'stagnes.website@gmx.fr';    //SMTP username
15     $mail->Password    = '!STAgne*2022';             //SMTP password
16     $mail->SMTPSecure  = PHPMailer::ENCRYPTION_SMTPS; //Enable implicit TLS encryption
17     $mail->Port        = 465;
18  ?>
```

Avec l'adresse e-mail GMX précédemment créée, les paramètres SMTP de GMX et le fichier « `phpmailer_connexion.php` », on va venir saisir les paramètres pour le serveur STMP à utiliser par PHPMailer :



Et enfin « `phpmailer_gestion.php` » qui va contenir les fonctions utilisant PHPMailer en « require » le fichier « `phpmailer_connexion` » à chaque début de fonction comme pour « `gestion_bdd` » et « `connexion.php` » :



Ajout des paramètres « setLanguage », « CharSet » et « Encoding » dans « phpmailer_connexion.php » pour s'assurer qu'il n'y ait pas de problème d'encodage et pour avoir les messages d'erreurs en français lorsque PHPMailer démarre, création de la variable « \$mail » et connexion au SMTP afin d'éviter de le faire pour chaque fonction :

```
includes > modeles > phpmailer_connexion.php
1  <?php
2      use PHPMailer\PHPMailer\PHPMailer;
3      use PHPMailer\PHPMailer\SMTP;
4      use PHPMailer\PHPMailer\Exception;
5
6      //Create an instance; passing 'true' enables exceptions
7      $mail = new PHPMailer(true);
8
9      //Server settings
10     $mail->SMTPDebug = SMTP::DEBUG_SERVER;           //Enable verbose debug output
11     $mail->isSMTP();                                 //Send using SMTP
12     $mail->Host = 'mail.gmx.com';                    //Set the SMTP server to send through
13     $mail->SMTPAuth = true;                          //Enable SMTP authentication
14     $mail->Username = 'stagnes.website@gmx.fr';      //SMTP username
15     $mail->Password = '!STAgne*2022';               //SMTP password
16     $mail->SMTPSecure = PHPMailer::ENCRYPTION_SMTPS; //Enable implicit TLS encryption
17     $mail->Port = 465;
18     $mail->setLanguage('fr', 'includes/lib/PHPMailer/language/'); //To load the French version
19     $mail->CharSet = 'UTF-8';
20     $mail->Encoding = 'base64';
21  ?>
```

Changement de « \$mail->SMTPDebug » à égal 0 afin de ne plus voir le Debugger de PHPMailer sur la page où il est utilisé :

```
includes > modeles > phpmailer_connexion.php
1  <?php
2      use PHPMailer\PHPMailer\PHPMailer;
3      use PHPMailer\PHPMailer\SMTP;
4      use PHPMailer\PHPMailer\Exception;
5
6      //Create an instance; passing 'true' enables exceptions
7      $mail = new PHPMailer(true);
8
9      //Server settings
10     $mail->SMTPDebug = 0; //Enable verbose debug output
11     $mail->isSMTP();      //Send using SMTP
12     $mail->Host = 'mail.gmx.com'; //Set the SMTP server to send through
13     $mail->SMTPAuth = true;      //Enable SMTP authentication
14     $mail->Username = 'stagnes.website@gmx.fr'; //SMTP username
15     $mail->Password = '!STAgne*2022'; //SMTP password
16     $mail->SMTPSecure = PHPMailer::ENCRYPTION_SMTPS; //Enable implicit TLS encryption
17     $mail->Port = 465;
18     $mail->setLanguage('fr', 'includes/lib/PHPMailer/language/'); //To load the French version
19     $mail->CharSet = 'UTF-8';
20     $mail->Encoding = 'base64';
21  ?>
```

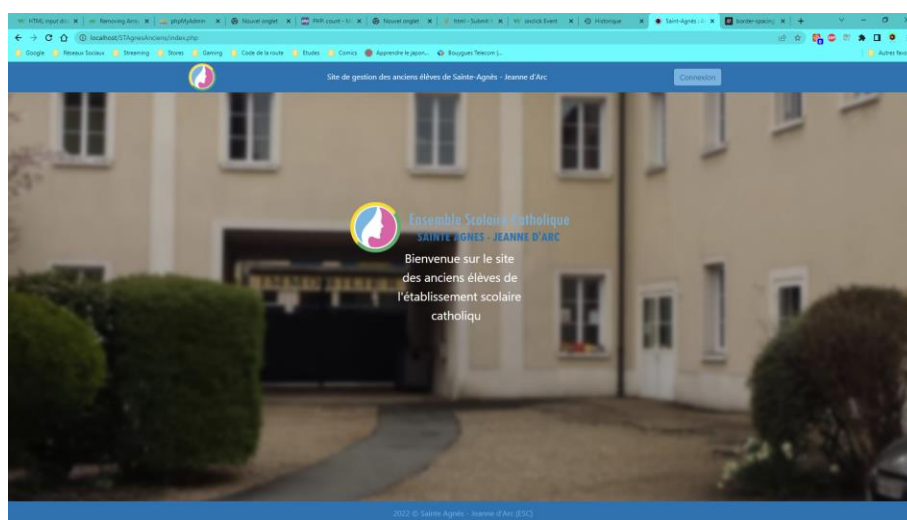
PHPMailer est installé et prêt à être utilisé. Il ne manque plus qu'à venir « include » dans l'ordre : « phpmailer_launcher.php » puis « phpmailer_connexion.php » et enfin « phpmailer_gestion.php » précédemment créés dans « index.php » afin de pouvoir l'utiliser partout sur l'application Web.

```
index.php
1  <?php
2      session_start();
3      include "includes/modeles/connexion.php";
4      include "includes/modeles/gestion_bdd.php";
5      include "includes/modeles/phpmailer_launcher.php";
6      include "includes/modeles/phpmailer_connexion.php";
7      include "includes/modeles/phpmailer_gestion.php";
8      include "controleurs/c_principal.php";
9  ?>
```

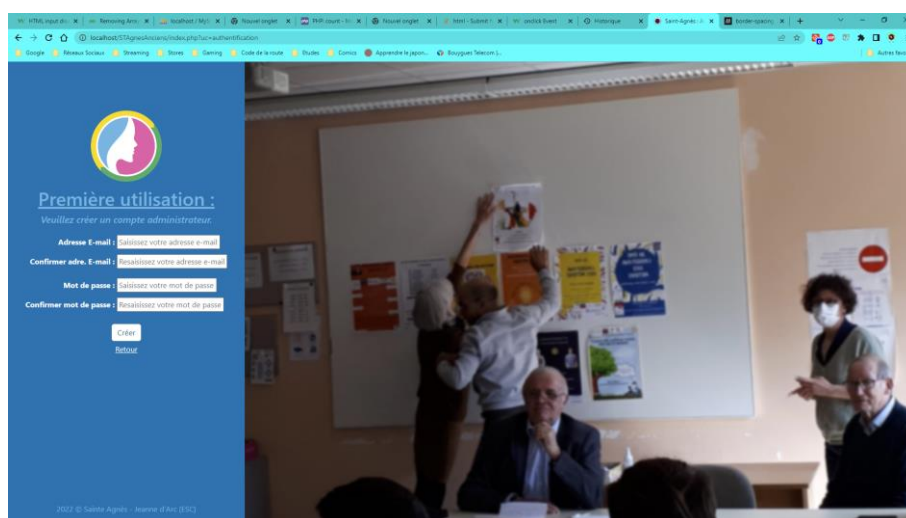
II) Application Web

2. Partie Authentification

La partie « authentification » est la première partie à laquelle aura accès l'utilisateur en se connectant à l'application Web. Cette partie est reliée à la page d'accueil tout en ayant son propre contrôleur et sa propre vue (Voir Annexe 2 : « Code du Contrôleur d'authentification » et Annexe 3 : « Code de la vue d'authentification »). Sur la page d'accueil, un bouton « Connexion » permet de se connecter s'il y a au moins un compte avec le droit « Administrateur ». Sinon, l'utilisateur sera amené sur une page « Première Utilisation » où il pourra inscrire le premier compte administrateur.

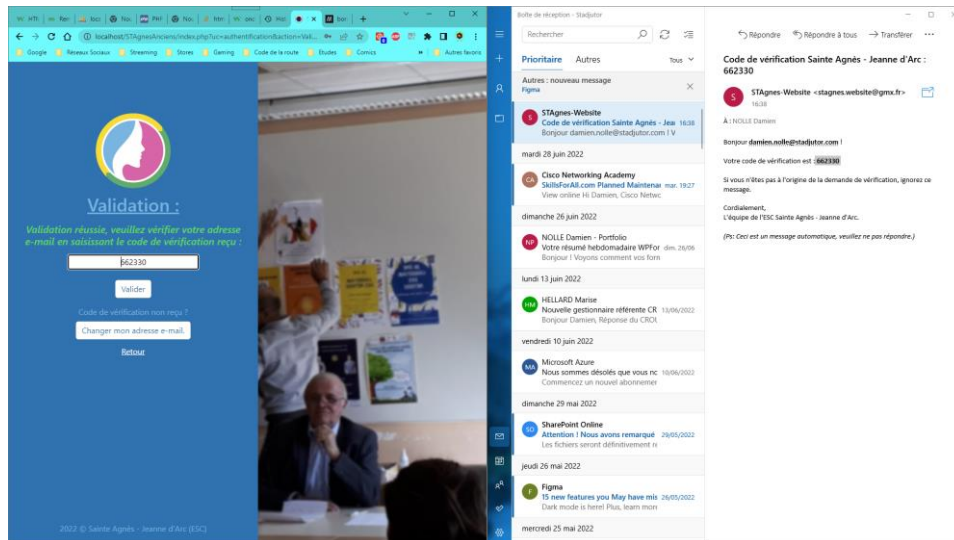


Page d'accueil avec le bouton « connexion » survolé par la souris de l'utilisateur.



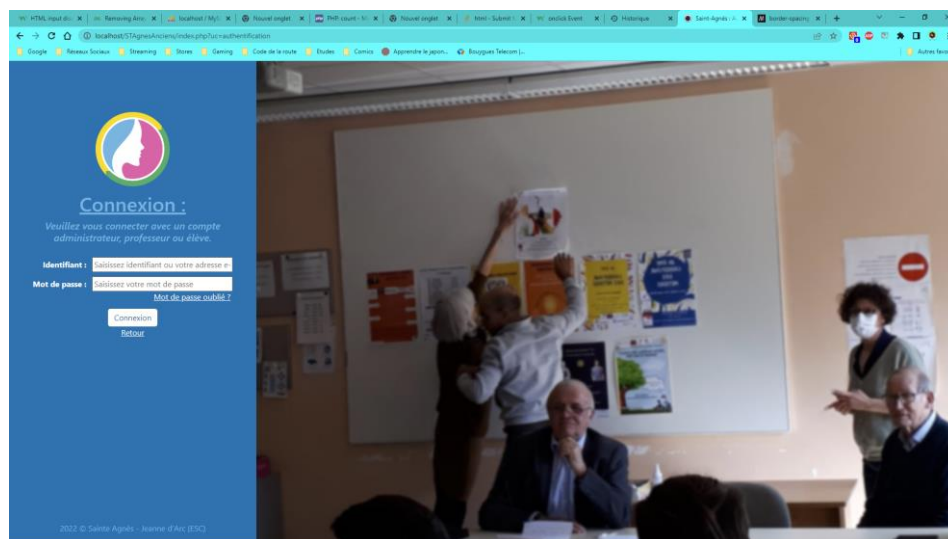
Page « Première Utilisation ».

Lors de cette inscription, l'utilisateur pourra entrer son adresse e-mail qui sera vérifiée par l'envoi d'un code qui devra être saisi sur l'application Web. Si l'utilisateur fait une erreur de saisie sur son adresse e-mail, il pourra la changer au cours de l'inscription. Une fois inscrit, l'utilisateur qui est maintenant administrateur sera redirigé vers la page d'accueil où il pourra se connecter en appuyant sur le bouton de connexion.



Validation d'un compte administrateur.

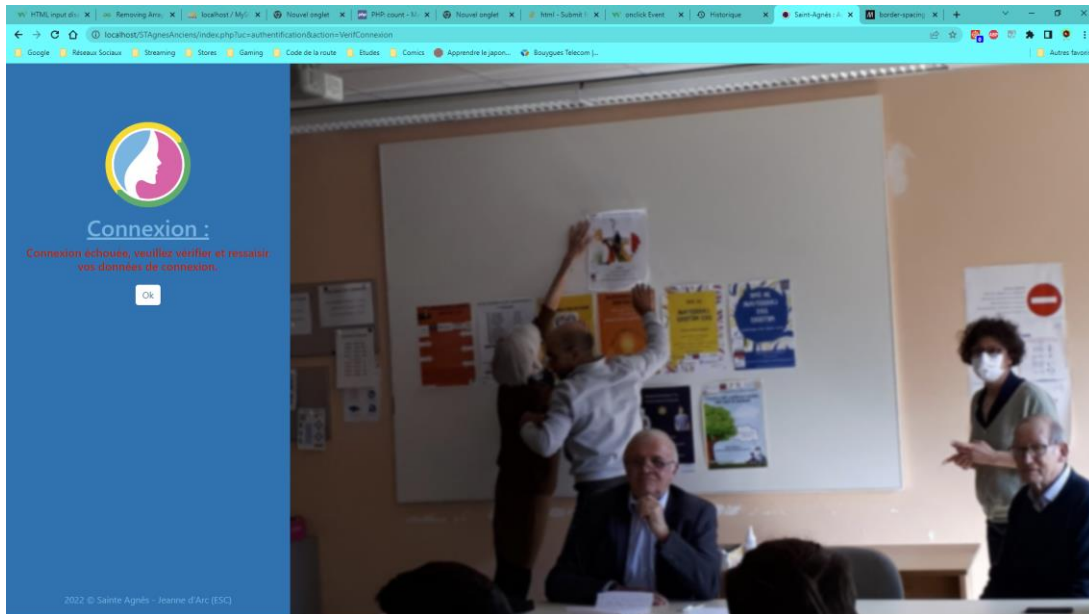
Une fois sur la page d'authentification, deux scénarios s'offrent à l'utilisateur : soit il entre son identifiant et son mot de passe puis se connecte, soit il a oublié son mot de passe et clique sur « Mot de passe oublié ? ». Si sur ce dernier scénario l'utilisateur est un administrateur, alors un code de vérification sera envoyé sur sa messagerie qu'il devra saisir afin de pouvoir changer son mot de passe. Si l'utilisateur est un professeur ou un élève, une page avec un message apparaît expliquant à l'utilisateur que au vu de son droit, il ne peut pas utiliser cette fonction mais qu'un mail a été envoyé à l'administrateur de l'application Web pour pouvoir changer le mot de passe.



Page de connexion.

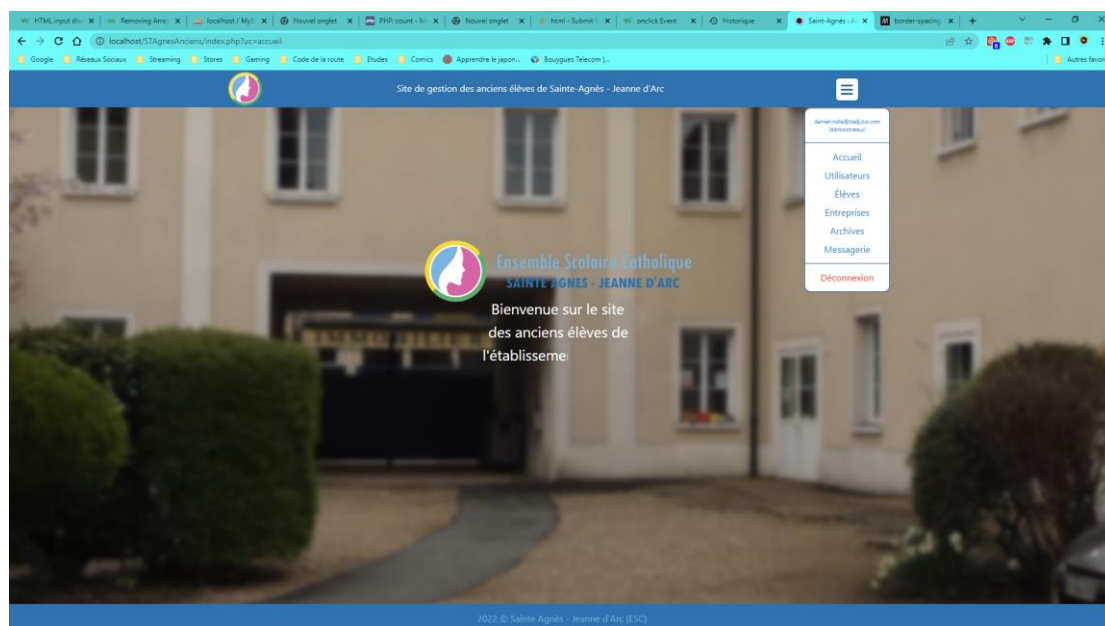
Toutes les données saisies sont contrôlées pour s'assurer du bon format et filtrer les attaques à l'injection SQL.

En cas d'erreur de contrôle des données, un message est affiché expliquant à l'utilisateur quelle erreur a pu être commise.



Exemple de message d'erreur.

Lorsque l'utilisateur est connecté, un menu lui permet d'accéder aux pages en fonction du droit accordé. Si l'utilisateur tente d'accéder à des pages qui ne lui sont pas accordées, il sera automatiquement redirigé vers l'accueil. Même chose s'il tente d'accéder à des pages alors qu'il n'est pas connecté.



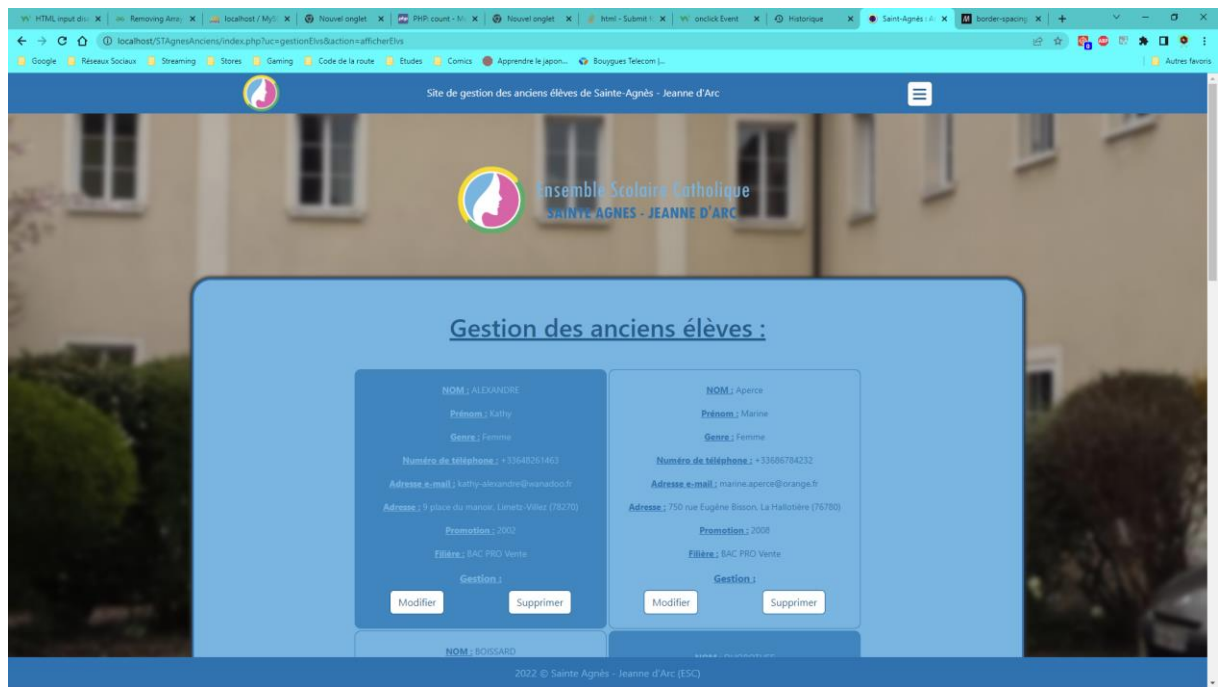
Le menu en question.

II) Application Web

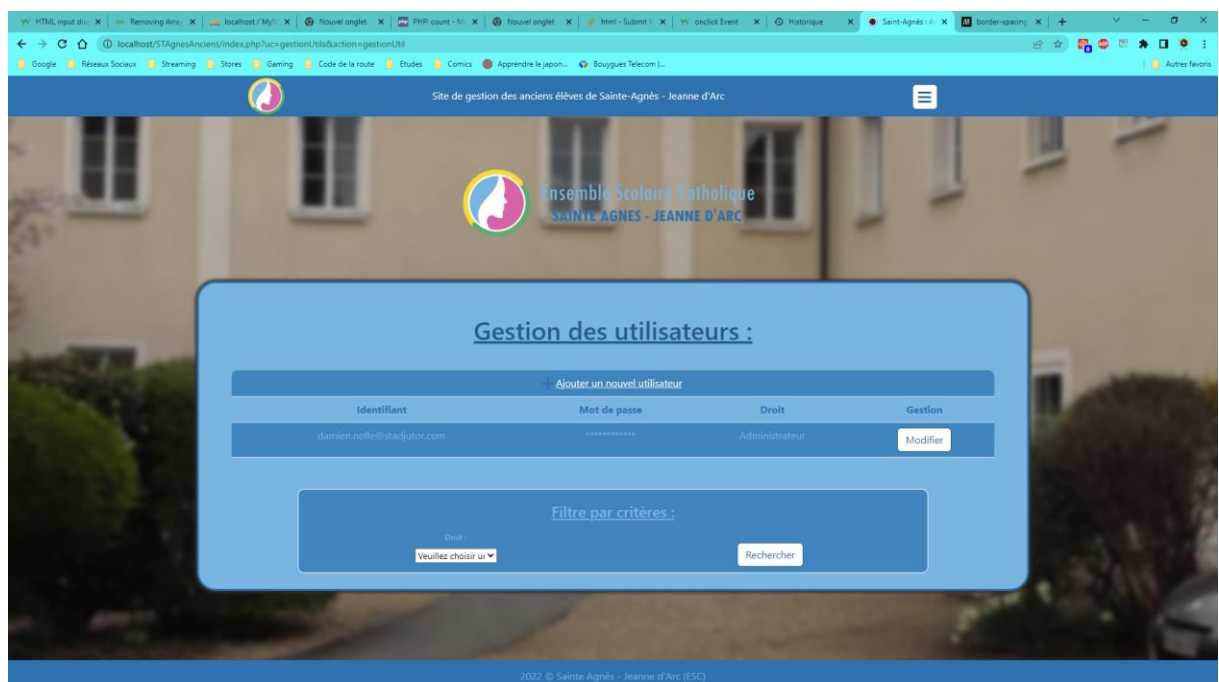
3. Partie Gestionnaire

La partie gestionnaire est composé de quatre gestionnaires :

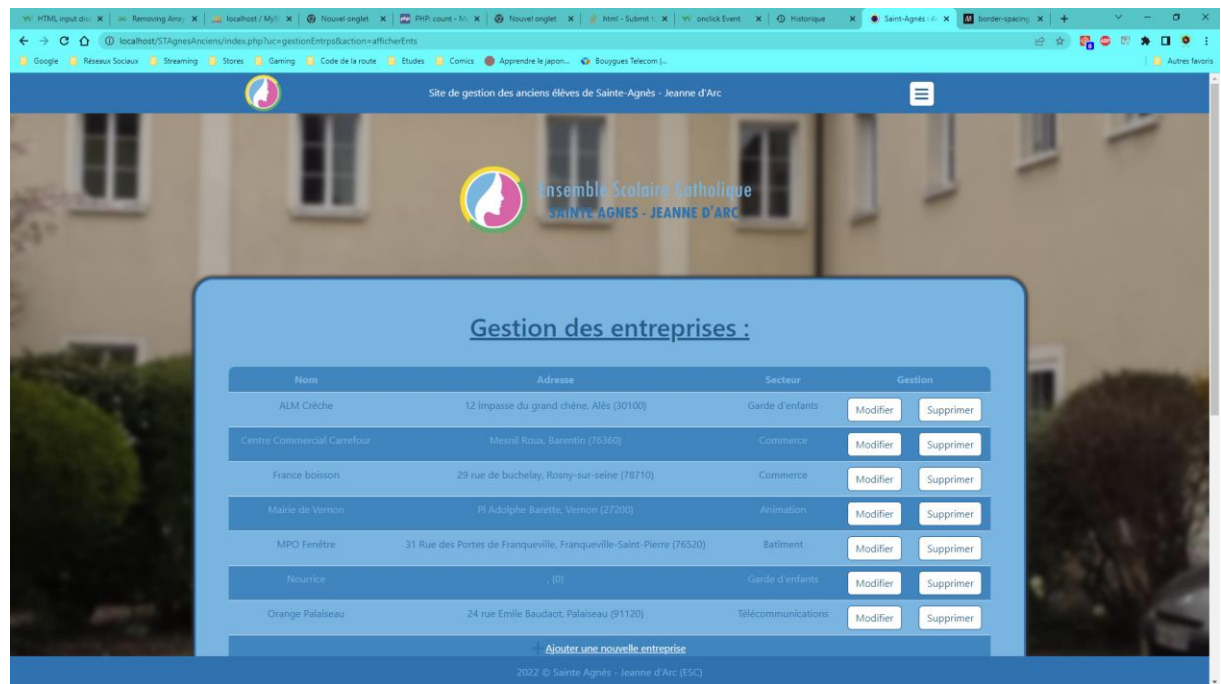
- Un gestionnaire pour les anciens élèves et leurs filières :



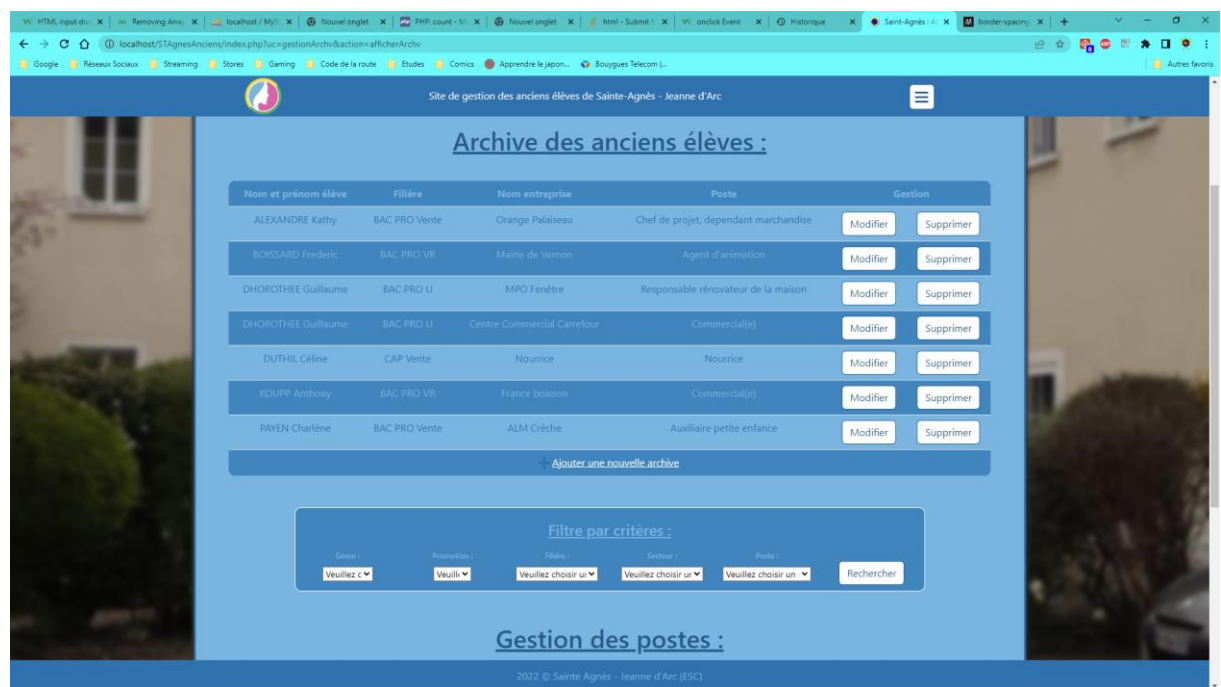
- Un Gestionnaire pour les utilisateurs :



- Un gestionnaire pour les entreprises et leurs secteurs :



- Et un gestionnaire « archive » qui est le résultat final faisant le lien entre les élèves, les entreprises et les postes :



Chaque gestionnaire permet de venir ajouter, modifier ou supprimer des données dans la base. Un filtre est présent sur chaque gestionnaire permettant de venir changer l'affichage des données en fonction de critères prédéfinis.

II) Application Web

4. Partie Messagerie

La messagerie est une messagerie classique permettant de venir saisir le titre, les destinataires et le contenu du message. Une comboBox permet de venir sélectionner les différents anciens élèves qui recevront le message et une autre comboBox permet de venir supprimer des destinataires de la liste. Une textBox permet de voir la liste entière des destinataires.

Un contrôle permet de s'assurer que au moins un destinataire est sélectionné et que le titre et le corps du message n'est pas vide. Si c'est le cas, un message erreur apparait sinon le message est envoyé.

The screenshot shows a web browser window with the URL `localhost/5TAnciens/index.php?uc=messagerie&action=afficherMessagerie`. The page has a blue header with the site name and a logo. The main content area features a 'Messagerie' form with the following fields:

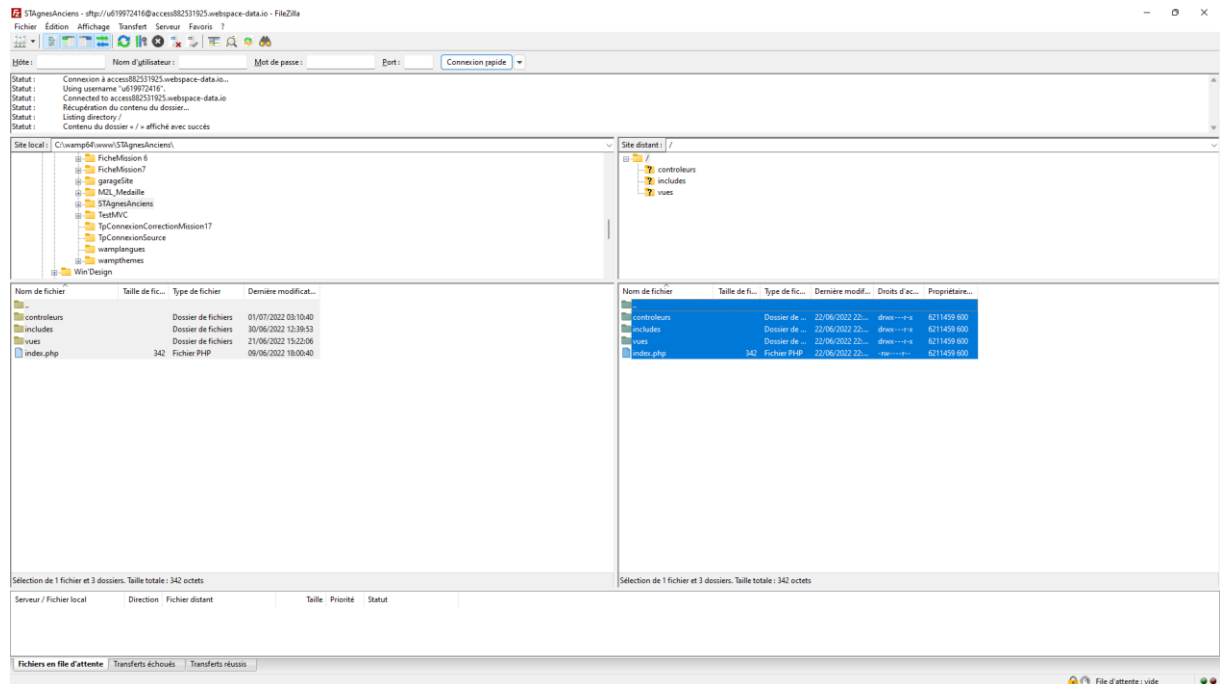
- Titre :** A text input field.
- Destinataire(s) :** A text input field.
- Actions:** Two buttons labeled 'Ajouter un destinataire' and 'Supprimer un destinataire'.
- Destinataire list:** A dropdown menu showing 'ALEXANDRE Kathy' and an 'Ajouter' button.
- Supprimer:** A dropdown menu and a 'Supprimer' button.
- Message :** A large text area for the message content.

The footer of the page indicates '2022 © Sainte-Agnès - Jeanne d'Arc (ESC)'.

La messagerie.

III) Hébergement

L'application Web et la base de données sont hébergées à l'établissement Saint-Adjutor. Un accès FTP m'a été confié, accessible via fileZila Client, afin d'assurer le transfert des fichiers et le bon fonctionnement ainsi que l'hébergement du site.



IV) Conclusion

Pour conclure, malheureusement, l'URL-Writing n'a pas pu être fait par manque de temps mais toute la partie fonctionnelle et graphique de l'application Web est terminée.

Il me n'était de ce fait pas possible de respecter totalement le cahier des charges par manque de temps.

Un problème d'organisation et de temps a été rencontré tout le long du stage, surtout à la fin. D'autant plus que certains problèmes étaient très difficiles à résoudre.

Malgré ces quelques problèmes, ces 5 semaines de stages m'ont permis de pratiquer et de revoir tout ce qui a été vu pendant l'année, que ça soit la rétro conception, le SQL ou encore le développement Web.

ANNEXES

I) Base de données

3. Modification

Annexe 1 : Importation des données dans les tables en requêtes SQL

- Importation des données dans la table des filières :

```
INSERT INTO tb_filiere  
(FLR_Libelle)  
VALUES ("CAP Vente");
```

```
INSERT INTO tb_filiere  
(FLR_Libelle)  
VALUES ("BAC PRO Vente");
```

```
INSERT INTO tb_filiere  
(FLR_Libelle)  
VALUES ("BAC PRO VR");
```

```
INSERT INTO tb_filiere  
(FLR_Libelle)  
VALUES ("BAC PRO U")
```

- **Importation des données dans la table des élèves :**

INSERT INTO tb_eleve

(ELV_Prenom, ELV_Nom, ELV_Genre, ELV_NumTel, ELV_Mail, ELV_Rue, ELV_Ville, ELV_CP,
ELV_AnneeLycee, ELV_FLR_Id)

VALUES ("Céline", "DUTHIL", 'F', "+33617803677", "celineduthil1983@gmail.com", "14 chemin du bois
de la croix", "La Chapelle des pots", 17000, 1999, 1);

INSERT INTO tb_eleve

(ELV_Prenom, ELV_Nom, ELV_Genre, ELV_NumTel, ELV_Mail, ELV_Rue, ELV_Ville, ELV_CP,
ELV_AnneeLycee, ELV_FLR_Id)

VALUES ("Frederic", "BOISSARD", 'H', "+33661930577", "fboissard@sna-vernon27.fr", "20 rue de
prevoyants appartement N°9 batiment 3 mat", "Le Grand Quevilly", 76120, 2002, 3);

INSERT INTO tb_eleve

(ELV_Prenom, ELV_Nom, ELV_Genre, ELV_NumTel, ELV_Mail, ELV_Rue, ELV_Ville, ELV_CP,
ELV_AnneeLycee, ELV_FLR_Id)

VALUES ("Charlène", "PAYEN", 'F', "+33666936709", "charlène.P74@hotmail.fr", "14 route canabias",
"Saint-Privat-des-Vieux", 30340, 2007, 2);

INSERT INTO tb_eleve

(ELV_Prenom, ELV_Nom, ELV_Genre, ELV_NumTel, ELV_Mail, ELV_Rue, ELV_Ville, ELV_CP,
ELV_AnneeLycee, ELV_FLR_Id)

VALUES ("Guillaume", "DHOROTHEE", 'H', "+33631740937", "guillaume.dorothe@hotmail.com", "36
chemin des lavenis", "Le vieil Evreux", 27930, 2001, 4);

INSERT INTO tb_eleve

(ELV_Prenom, ELV_Nom, ELV_Genre, ELV_NumTel, ELV_Mail, ELV_Rue, ELV_Ville, ELV_CP,
ELV_AnneeLycee, ELV_FLR_Id)

VALUES ("Kathy", "ALEXANDRE", 'F', "+33648261463", "kathy-alexandre@wanadoo.fr", "9 place du
manoir", "Limetz-Ville", 78270, 2002, 2);

INSERT INTO tb_eleve

(ELV_Prenom, ELV_Nom, ELV_Genre, ELV_NumTel, ELV_Mail, ELV_Rue, ELV_Ville, ELV_CP,
ELV_AnneeLycee, ELV_FLR_Id)

```
VALUES ("Anthony","KOUUP",'H'," +33671646133","enzo_lola.ak@gmail.com","1 rue Claude Monet",  
"Pacy-sur-Eure",27120,2001,3);
```

```
INSERT INTO tb_eleve
```

```
(ELV_Prenom, ELV_Nom, ELV_Genre, ELV_NumTel, ELV_Mail, ELV_Rue, ELV_Ville, ELV_CP,  
ELV_AnneeLycee, ELV_FLR_Id)
```

```
VALUES ("Marine","Aperce",'F'," +33686784232","marine.aperce@orange.fr","750 rue Eugène  
Bisson", "La Hallotière",76780,2008,2)
```

- **Importation des données dans la table des secteurs :**

```
INSERT INTO tb_secteur
```

```
(SCT_Libelle)
```

```
VALUES ("Garde d'enfants");
```

```
INSERT INTO tb_secteur
```

```
(SCT_Libelle)
```

```
VALUES ("Animation");
```

```
INSERT INTO tb_secteur
```

```
(SCT_Libelle)
```

```
VALUES ("Bâtiment");
```

```
INSERT INTO tb_secteur
```

```
(SCT_Libelle)
```

```
VALUES ("Télécommunications");
```

```
INSERT INTO tb_secteur
```

```
(SCT_Libelle)
```

```
VALUES ("Commerce")
```

- **Importation des données dans la table des entreprises :**

```
INSERT INTO tb_entreprise
```

```
(ENT_Nom, ENT_SCT_Id)
```

```
VALUES ("Nourrice", 1);
```

```
INSERT INTO tb_entreprise
```

```
(ENT_Nom, ENT_Rue, ENT_Ville, ENT_CP, ENT_SCT_Id)
```

```
VALUES ("Mairie de Vernon", "Pl Adolphe Barette", "Vernon", 27200, 2);
```

```
INSERT INTO tb_entreprise
```

```
(ENT_Nom, ENT_Rue, ENT_Ville, ENT_CP, ENT_SCT_Id)
```

```
VALUES ("ALM Crèche", "12 impasse du grand chène", "Alès", 30100, 1);
```

```
INSERT INTO tb_entreprise
```

```
(ENT_Nom, ENT_Rue, ENT_Ville, ENT_CP, ENT_SCT_Id)
```

```
VALUES ("MPO Fenêtre", "31 Rue des Portes de Franqueville", "Franqueville-Saint-Pierre", 76520, 3);
```

```
INSERT INTO tb_entreprise
```

```
(ENT_Nom, ENT_Rue, ENT_Ville, ENT_CP, ENT_SCT_Id)
```

```
VALUES ("Orange Palaiseau", "24 rue Emile Baudaot", "Palaiseau", 91120, 4);
```

```
INSERT INTO tb_entreprise
```

```
(ENT_Nom, ENT_Rue, ENT_Ville, ENT_CP, ENT_SCT_Id)
```

```
VALUES ("France boisson", "29 rue de buchelay", "Rosny-sur-seine", 78710, 5);
```

```
INSERT INTO tb_entreprise
```

```
(ENT_Nom, ENT_Rue, ENT_Ville, ENT_CP, ENT_SCT_Id)
```

```
VALUES ("Centre Commercial Carrefour", "Mesnil Roux", "Barentin", 76360, 5)
```

- **Importation des données dans la table des postes :**

```
INSERT INTO tb_poste
```

```
(PST_Libelle)
```

```
VALUES ("Nourrice");
```

```
INSERT INTO tb_poste
```

```
(PST_Libelle)
```

```
VALUES ("Agent d'animation");
```

```
INSERT INTO tb_poste
```

```
(PST_Libelle)
```

```
VALUES ("Auxiliaire petite enfance");
```

```
INSERT INTO tb_poste
```

```
(PST_Libelle)
```

```
VALUES ("Responsable rénovateur de la maison");
```

```
INSERT INTO tb_poste
```

```
(PST_Libelle)
```

```
VALUES ("Chef de projet, dependant marchandise");
```

```
INSERT INTO tb_poste
```

```
(PST_Libelle)
```

```
VALUES ("Commercial(e)");
```

- **Importation des données dans la table « travailler » :**

```
INSERT INTO tb_travailler  
(TRV_ENT_Id, TRV_ELV_Id, TRV_PST_Id)  
VALUES(1,1,1);
```

```
INSERT INTO tb_travailler  
(TRV_ENT_Id, TRV_ELV_Id, TRV_PST_Id)  
VALUES(2,2,2);
```

```
INSERT INTO tb_travailler  
(TRV_ENT_Id, TRV_ELV_Id, TRV_PST_Id)  
VALUES(3,3,3);
```

```
INSERT INTO tb_travailler  
(TRV_ENT_Id, TRV_ELV_Id, TRV_PST_Id)  
VALUES(4,4,4);
```

```
INSERT INTO tb_travailler  
(TRV_ENT_Id, TRV_ELV_Id, TRV_PST_Id)  
VALUES(5,5,5);
```

```
INSERT INTO tb_travailler  
(TRV_ENT_Id, TRV_ELV_Id, TRV_PST_Id)  
VALUES(6,6,6);
```

```
INSERT INTO tb_travailler  
(TRV_ENT_Id, TRV_ELV_Id, TRV_PST_Id)  
VALUES(7,4,6)
```


- **Importation des données dans la table des droits :**

```
INSERT INTO tb_droits  
(DRT_Libelle)  
VALUES("Administrateur");
```

```
INSERT INTO tb_droits  
(DRT_Libelle)  
VALUES("Professeur");
```

```
INSERT INTO tb_droits  
(DRT_Libelle)  
VALUES("Élève")
```

NB : La table utilisateur sera rempli un peu plus tard dans l'avancement du projet en raison de la génération des mots de passes cryptés.

II) Application Web

2. Partie Authentification

Annexe 2 : Code du Contrôleur d'authentification

```
<?php
if (isset($_SESSION['validCoDroit']) && isset($_SESSION['validCoIdentifiant']))
{
    header("Location:index.php");
}
else
{
    if (!isset($_REQUEST['action']))
    {
        $action = "afficherAuth" ;
    }
    else
    {
        $action = $_REQUEST['action'] ;
    }

    switch ($action)
    {
        case "afficherAuth" :
        {
            $_SESSION['nbEnvoiPHPMailer'] = 0;
            $_SESSION['enregMDPBase'] = 0;
            $_SESSION['codeVerifLimite'] = 0;

            $existeUtil = verifExisteUtil();

            if ($existeUtil == false)
            {
                if (isset($_SESSION['codeVerifLimite1erUtil']))
                {
                    $_SESSION['codeVerifLimite1erUtil'] = 0;
                }
                else
                {
                    header("Location:index.php");
                }

                if (isset($_SESSION['codeVerifLimiteChangerEmail']))
                {
                    $_SESSION['codeVerifLimiteChangerEmail'] = 0;
                }
                else
                {
```

```

        header("Location:index.php");
    }
}

if (isset($_POST['changerEmailMDPValid']))
{
    $changerEmailMDP = $_POST['changerEmailMDPValid'];
}

if (isset($_POST['emailOriginal']))
{
    $changerEmailOriginal = $_POST['emailOriginal'];
}

if (isset($_POST['codeVerifOriginal']))
{
    $changerEmailVerifCodeOriginal = $_POST['codeVerifOriginal'];
}

if (isset($_POST['changerEmailValid']))
{
    $comfirmerChangerEmailValid = $_POST['changerEmailValid'];
}

if (isset($_POST['codeVerifSaisiEchoue']))
{
    $comfirmerCodeSaisieEchoue = $_POST['codeVerifSaisiEchoue'];
}

require "vues/v_authentification.php" ;
break ;
}

case "VerifConnexion" :
{
    if (isset($_POST['identifiantConnexion']) && (isset($_POST['mdpConnexion'])))
    {
        $verifIdentifiant = $_POST['identifiantConnexion'];
        $verifMDP = $_POST['mdpConnexion'];
        $validConnexion = false;

        if (verifIDMDP($verifIdentifiant,$verifMDP) != null)
        {
            $_SESSION['validCoDroit'] = verifIDMDP($verifIdentifiant,$verifMDP);
            $_SESSION['validCoIdentifiant'] = $verifIdentifiant;
            $validConnexion = true;
        }

        require "vues/v_authentification.php" ;
    }
}

```

```

        break;
    }
    else
    {
        header("Location:index.php");
    }
}

case "MDPOublie" :
{
    if (isset($_SESSION['codeVerifLimite']))
    {
        if (isset($_POST['identifiantMDPOublie']) && !isset($_POST['codeVerifMDPOublieSaisi']) &&
!isset($_POST['changerMDP']) && !isset($_POST['confirmerChangerMDP']) && !isset($_POST['validCodeMDPOublie']))
        {
            $idMDPOublieVerifValid = verifIDOublie($_POST['identifiantMDPOublie']);

            if ($idMDPOublieVerifValid == "Administrateur")
            {
                $EmailMDPOublie = $_POST['identifiantMDPOublie'];

                if ($_SESSION['codeVerifLimite'] == 0)
                {
                    $codeVerifEmailMDPOublie = rand(100000, 999999);

                    $_SESSION['codeVerifLimite'] = $codeVerifEmailMDPOublie;

                    EnvoiCode($EmailMDPOublie,$codeVerifEmailMDPOublie);
                }
                else
                {
                    $codeVerifEmailMDPOublie = $_SESSION['codeVerifLimite'];
                }
            }

            if ($idMDPOublieVerifValid == "Élève" || $idMDPOublieVerifValid == "Professeur")
            {
                $IDMDPOublie = $_POST['identifiantMDPOublie'];

                if ($_SESSION['nbEnvoiPHPMailer'] == 0)
                {
                    EnvoiMessageAdminMDPOublie($IDMDPOublie,$idMDPOublieVerifValid);
                    $_SESSION['nbEnvoiPHPMailer']++;
                }
            }
        }
    }
}

```

```

        if (isset($_POST['identifiantMDPOublie']) && isset($_POST['codeVerifMDPOublieSaisi']) &&
!isset($_POST['changerMDP']) && !isset($_POST['confirmerChangerMDP']) && !isset($_POST['validCodeMDPOublie']))
        {
            $idMDPOublieVerifValid = $_POST['DroitMDPOublie'];
            $EmailMDPOublie = $_POST['identifiantMDPOublie'];
            $codeVerifEmailMDPOublie = $_POST['codeVerifMDPOublie'];
            $codeVerifSaisiEmailMDPOublie = $_POST['codeVerifMDPOublieSaisi'];
            $validCodeMDPOublie = false;

            $verifCodeMDPOublie = verifCode($codeVerifEmailMDPOublie);
            $verifSaisiCodeMDPOublie = verifCode($codeVerifSaisiEmailMDPOublie);

            if ($verifSaisiCodeMDPOublie === true && $verifCodeMDPOublie === true)
            {
                if ($codeVerifEmailMDPOublie == $codeVerifSaisiEmailMDPOublie)
                {
                    $validCodeMDPOublie = true;
                }
            }
        }

        if (!isset($_POST['changerMDP']) && !isset($_POST['confirmerChangerMDP']) &&
isset($_POST['validCodeMDPOublie']))
        {
            $idMDPOublieVerifValid = $_POST['identifiantMDPOublie'];
            $validCodeMDPOublie = $_POST['validCodeMDPOublie'];
            $EmailMDPOublie = $_POST['emailMDPOublie'];
            $codeVerifSaisiEmailMDPOublie = $_POST['codeVerifMDPOublieSaisi'];
        }

        if (isset($_POST['changerMDP']) && isset($_POST['confirmerChangerMDP']))
        {
            $etatChangementMDP = false;
            $idMDPOublieVerifValid = $_POST['identifiantMDPOublie'];
            $validCodeMDPOublie = $_POST['validCodeMDPOublie'];
            $EmailMDPOublie = $_POST['emailMDPOublie'];
            $codeVerifSaisiEmailMDPOublie = $_POST['codeVerifMDPOublieSaisi'];

            if ($_POST['changerMDP'] == $_POST['confirmerChangerMDP'])
            {
                $verifValidChangerMDP = verifMDP($_POST['changerMDP']);

                if ($verifValidChangerMDP === true)
                {
                    $etatChangementMDP = true;

                    if ($_SESSION['enregMDPBase'] == 0)
                    {
                        $temps = 0.1;

```

```

        $cost = 10;

        do
        {
            $debut = microtime(true);
            password_hash($_POST['changerMDP'], PASSWORD_DEFAULT, ['cost' => $cost]);
            $fin = microtime(true);
            $cost++;
        }
        while (($fin - $debut) < $temps);

        $validChangerMDP = password_hash($_POST['changerMDP'], PASSWORD_DEFAULT, ['cost'
=> $cost]);

        changerMDPUtil($EmailMDPOublie,$idMDPOublieVerifValid,$validChangerMDP);

        $_SESSION['enregMDPBase']++;
    }
}
}

require "vues/v_authentification.php" ;
break;
}
else
{
    header("Location:index.php");
}
}

case "ValidlerUtil" :
{
    if (!isset($_SESSION['codeVerifLimite1erUtil']) && !isset($_SESSION['codeVerifLimiteChangerEmail'])
|| !isset($_POST['Email1erUtil']) && !isset($_POST['Email1erUtilVerif']) && !isset($_POST['MDP1erUtil']) &&
!isset($_POST['MDP1erUtilVerif']) && !isset($_POST['codeVerifEmailSaisi']))
    {
        header("Location:index.php");
    }
    else
    {
        $verifCodeEgal = false;
        $cptEtatValid = 0;

        if (!isset($_POST['authChangerEmailMDP']) && !isset($_POST['retourMDP']) &&
!isset($_POST['retourEmail']) && !isset($_POST['retourCodeVerif']) && !isset($_POST['codeVerifEmailSaisi']) &&
!isset($_POST['validCodeVerifSaisi']) && !isset($_POST['validEmailSaisi']) && !isset($_POST['validMDPSaisi']) &&
!isset($_POST['erreurRenvoiOriginalEmail']) && !isset($_POST['erreurRenvoiOriginalCodeVerif']))

```

```

{
    if ($_POST['EmaillierUtil'] == $_POST['EmaillierUtilVerif'])
    {
        $cptEtatValid++;

        if ($_POST['MDP1erUtil'] == $_POST['MDP1erUtilVerif'])
        {
            $cptEtatValid++;

            if ($_POST['EmaillierUtil'] != $_POST['MDP1erUtil'])
            {
                $cptEtatValid++;

                $verifValidEmail = verifEmail($_POST['EmaillierUtil']);
                $verifValidMDP = verifMDP($_POST['MDP1erUtil']);

                if ($verifValidEmail === true)
                {
                    $cptEtatValid++;

                    if ($verifValidMDP === true)
                    {
                        $cptEtatValid++;

                        $temps = 0.1;
                        $cost = 10;

                        do
                        {
                            $debut = microtime(true);
                            password_hash($_POST['MDP1erUtil'], PASSWORD_DEFAULT, ['cost' =>
$cost]);

                            $fin = microtime(true);
                            $cost++;
                        }
                        while (($fin - $debut) < $temps);

                        $validMDP = password_hash($_POST['MDP1erUtil'], PASSWORD_DEFAULT, ['cost'
=> $cost]);

                        $validEmail = $_POST['EmaillierUtil'];

                        if ($_SESSION['codeVerifLimite1erUtil'] == 0)
                        {
                            $codeVerifEmail = rand(100000, 999999);

                            $_SESSION['codeVerifLimite1erUtil'] = $codeVerifEmail;

                            EnvoiCode($validEmail,$codeVerifEmail);

```

```

        }
        else
        {
            $codeVerifEmail = $_SESSION['codeVerifLimite1erUtil'];
        }
    }
}
}
}
}
}

if (isset($_POST['authChangerEmailMDP']) && !isset($_POST['retourMDP']) &&
!isset($_POST['retourEmail']) && !isset($_POST['retourCodeVerif']))
{
    $validMDP = $_POST['authChangerEmailMDP'];

    if ($_POST['Email1erUtil'] == $_POST['Email1erUtilVerif'])
    {
        $verifChangerValidEmail = verifEmail($_POST['Email1erUtil']);

        $cptEtatValid = 3;

        if ($verifChangerValidEmail === true)
        {
            $cptEtatValid = 5;

            $validEmail = $_POST['Email1erUtil'];

            if ($_SESSION['codeVerifLimiteChangerEmail'] == 0)
            {
                $codeVerifEmail = rand(100000, 999999);

                $_SESSION['codeVerifLimiteChangerEmail'] = $codeVerifEmail;

                EnvoiCode($validEmail,$codeVerifEmail);
            }
            else
            {
                $codeVerifEmail = $_SESSION['codeVerifLimiteChangerEmail'];
            }
        }
    }
}

if (isset($_POST['erreurRenvoiOriginalEmail']) && isset($_POST['erreurRenvoiOriginalCodeVerif']))
{
    $erreurValidEmail = $_POST['erreurRenvoiOriginalEmail'];
    $erreurValidCodeVerif = $_POST['erreurRenvoiOriginalCodeVerif'];
}

```



```

    }

    if (isset($_POST['retourMDP']) && isset($_POST['retourEmail']) &&
isset($_POST['retourCodeVerif']))
    {
        $cptEtatValid = 5;
        $validMDP = $_POST['retourMDP'];
        $validEmail = $_POST['retourEmail'];
        $codeVerifEmail = $_POST['retourCodeVerif'];
    }

    if (isset($_POST['codeVerifEmailSaisi']) && isset($_POST['validCodeVerifSaisi']) &&
isset($_POST['validEmailSaisi']) && isset($_POST['validMDPSaisi']))
    {
        $cptEtatValid = 5;
        $codeVerifEmailSaisi = $_POST['codeVerifEmailSaisi'];
        $validMDP = $_POST['validMDPSaisi'];
        $validEmail = $_POST['validEmailSaisi'];
        $codeVerifEmail = $_POST['validCodeVerifSaisi'];

        if ($codeVerifEmailSaisi == $codeVerifEmail)
        {
            $verifCodeVerifEmailSaisi = verifCode($codeVerifEmailSaisi);
            $verifCodeVerifEmail = verifCode($codeVerifEmail);

            if ($verifCodeVerifEmailSaisi === true && $verifCodeVerifEmail === true)
            {
                $verifCodeEgal = true;

                if (verifEmailExiste($validEmail) == false)
                {
                    //$temps = 0.1;
                    //$cost = 10;

                    //do
                    //{
                        // $debut = microtime(true);
                        // password_hash($validEmail, PASSWORD_DEFAULT, ['cost' => $cost]);
                        // $fin = microtime(true);
                        // $cost++;
                    //}
                    //while (($fin - $debut) < $temps);

                    //$validEmail = password_hash($validEmail, PASSWORD_DEFAULT, ['cost' => $cost]);

                    enregUtilAdmin($validEmail, $validMDP);
                }
            }
        }
    }
}

```

```
    }

    require "vues/v_authentication.php" ;
    break;
  }
}
}
}
?>
```

Annexe 3 : Code de la vue d'authentification

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Saint-Agnès : Authentication</title>

    <!-- favicon -->
    <link rel="icon" type="image/png" href="includes/img/logo_favicon.png" />

    <!-- CSS -->
    <link rel="stylesheet" href="includes/css/bootstrap.css">
    <link rel="stylesheet" href="includes/css/style-auth.css">

    <!-- Js -->
    <script src="includes/js/fonctions.js"></script>
  </head>

  <body draggable="false" ondragstart="return false;">
    <div class="container-fluid auth-corps-bg" draggable="false" ondragstart="return false;">
      <div class="container auth-contenu" draggable="false" ondragstart="return false;">
        <div class="row auth-contenu-ligne">
          <div class="col auth-contenu-colonne">
            

            <?php
              if (!isset($_REQUEST['action']) || isset($_REQUEST['action']) && $_REQUEST['action'] == "afficherAuth")
              {
                if ($existeUtil == false)
                {
                  if (!isset($_POST['changerEmailMDPValid']))
                  {
                    echo '<div class="auth-contenu-titres">
                      <h1>Première utilisation :</h1>
                      <h5>Veuillez créer un compte administrateur.</h5>
                    </div>';
                  }
                }
                else
                {
                  echo '<div class="auth-contenu-titres">
                    <h1>Changer adresse e-mail :</h1>
                    <h5>Veuillez saisir la nouvelle adresse e-mail.</h5>
                  </div>';
                }
              }

              echo '<div class = "auth-contenu-form">
                <form name="auth-form" id="auth-form" method="POST"
action="index.php?uc=authentification&action=ValidierUtil">
```

```

        <table>

            <tr class="auth-contenu-form-table-ecartemail">

                <th>Adresse E-mail :</th>

                <td><input name="Email1erUtil" type="email" id="Email1erUtil" placeholder="Saisissez
votre adresse e-mail" size="24" maxlength="40"></td>

            </tr>

            <tr class="auth-contenu-form-table-ecart">

                <th>Confirmer adre. E-mail :</th>

                <td><input name="Email1erUtilVerif" type="email" id="Email1erUtilVerif"
placeholder="Resaisissez votre adresse e-mail" size="26" maxlength="40"></td>

            </tr>';

        if (!isset($_POST['changerEmailMDPValid']))
        {
            echo '<tr class="auth-contenu-form-table-ecart">

                <th>Mot de passe :</th>

                <td><input name="MDP1erUtil" type="password" id="MDP1erUtil" placeholder="Saisissez
votre mot de passe" size="23" maxlength="25"></td>

            </tr>

            <tr class="auth-contenu-form-table-ecartmdp">

                <th>Confirmer mot de passe :</th>

                <td><input name="MDP1erUtilVerif" type="password" id="MDP1erUtilVerif"
placeholder="Resaisissez votre mot de passe" size="25" maxlength="25"></td>

            </tr>

            <tr>

                <td colspan="2"><button type="submit" class="btn btn-primary
creer">Cr  er</button></td>

            </tr>';
        }
        else
        {
            echo '<tr>

                <td colspan="2">

                    <button type="submit" class="btn btn-primary btn-changeremail">Changer adresse e-
mail</button>

                    <input type="hidden" name="authChangerEmailMDP" id="authChangerEmailMDP"
value="'. $changerEmailMDP. '">

                    <input type="hidden" name="erreurRenvoiOriginalEmail"
id="erreurRenvoiOriginalEmail" value="'. $changerEmailOriginal. '">

                    <input type="hidden" name="erreurRenvoiOriginalCodeVerif"
id="erreurRenvoiOriginalCodeVerif" value="'. $changerEmailVerifCodeOriginal. '">';

                </td></td>

                <td colspan="2">

                    <input type="hidden" name="retourConfirmationErreur"
id="retourConfirmationErreur" value="'. $comfirmerChangerEmailValid. '">';

                </td></td>

            </tr>

            <tr>

                <td colspan="2">

                    <input type="hidden" name="codeVerifSaisiEchoue" id="codeVerifSaisiEchoue" value="'. $codeVerifSaisiEchoue. '">';

                </td></td>

            </tr>';
        }
    }
}

if (isset($_POST['codeVerifSaisiEchoue']))
{
    // ...
}

```

```

        {
            echo '<input type="hidden" name="retourConfirmationCodeSaisiEchoueErreur"
id="retourConfirmationCodeSaisiEchoueErreur" value="'. $comfirmerCodeSaisieEchoue. '">';
        }

        echo '</td>

        </tr>';
    }

    echo '
    </table>

    </form>

    </div>';

    if (!isset($_POST['changerEmailMDPValid']))
    {
        echo '<div class="auth-contenu-retour">

        <table>

        <tr>

        <td><a href="index.php?uc=accueil">Retour</a></td>

        </tr>

        </table>

        </div>';
    }
    else
    {
        echo '<div class="auth-contenu-retour">

        <table>

        <tr>

        <td>

        <form name="retour-changeremail" id="retour-changeremail" method="POST"
action="index.php?uc=authentification&action=ValidierUtil">

        <a href="javascript:retourchangeremail()">Retour</a>

        <input type="hidden" name="retourMDP" id="retourMDP"

value="'. $changerEmailMDP. '">

        <input type="hidden" name="retourEmail" id="retourEmail"

value="'. $changerEmailOriginal. '">

        <input type="hidden" name="retourCodeVerif" id="retourCodeVerif"

value="'. $changerEmailVerifCodeOriginal. '">';

        if (isset($_POST['changerEmailValid']))
        {
            echo '<input type="hidden" name="retourConfirmation"
id="retourConfirmation" value="'. $comfirmerChangerEmailValid. '">';
        }

        if (isset($_POST['codeVerifSaisiEchoue']))
        {

```

```

        echo ' <input type="hidden" name="retourConfirmationCodeSaisiEchoue"
id="retourConfirmationCodeSaisiEchoue" value="'. $comfirmerCodeSaisieEchoue. ' ">';
    }

    echo ' </form>
</td>
</tr>
</table>
</div>';
}
}
else
{
    echo ' <div class="auth-contenu-titres">
<h1>Connexion :</h1>
<h5>Veuillez vous connecter avec un compte administrateur, professeur ou élève.</h5>
</div>

<div class="auth-contenu-form-connexion">
<form method="POST" action="index.php?uc=authentification&action=VerifConnexion">
<table class="table-connexion">
<tr class="auth-contenu-form-table-ecartConnexion">
<th>Identifiant :</th>
<td><input name="identifiantConnexion" type="text" placeholder="Saisissez identifiant
ou votre adresse e-mail" id="identifiantConnexion" size="33" maxlength="40"></td>
</tr>
<tr class="auth-contenu-form-table-ecartConnexion">
<th>Mot de passe :</th>
<td><input name="mdpConnexion" type="password" placeholder="Saisissez votre mot de
passe" id="mdpConnexion" size="33" maxlength="25"></td>
</tr>
<tr>
<td colspan="2"><div class="a-mdp-oublie"><a
href="index.php?uc=authentification&action=MDPOublie">Mot de passe oublié ?</a></div></td>
</tr>
<tr>
<td colspan="2"><button type="submit" class="btn btn-primary
connexion">Connexion</button></td>
</tr>
</table>
</form>
</div>

<div class="auth-contenu-retour-connexion">
<table>
<tr>
<td><a href="index.php?uc=accueil">Retour</a></td>
</tr>
</table>

```

```

        </div>';
    }
}

if (isset($_REQUEST['action']) && $_REQUEST['action'] == "VerifConnexion")
{
    if ($validConnexion == true)
    {
        echo '<div class="auth-contenu-titres-connexionreussi">
            <h1>Connexion :</h1>
            <h5>Connexion réussie au compte : '.$_SESSION['validCoIdentifiant'].'
('.$_SESSION['validCoDroit'].').</h5>
        </div>

        <form method="POST" action="index.php?uc=accueil"><button type="submit" class="btn btn-primary
ok">Ok</button></form>';
    }
    else
    {
        echo '<div class="auth-contenu-titres-connexionechoue">
            <h1>Connexion :</h1>
            <h5>Connexion échouée, veuillez vérifier et ressaisir vos données de connexion.</h5>
        </div>

        <form method="POST" action="index.php?uc=authentification"><button type="submit" class="btn btn-
primary ok">Ok</button></form>';
    }
}

if (isset($_REQUEST['action']) && $_REQUEST['action'] == "MDPOublie")
{
    echo '<div class="auth-contenu-titres">
        <h1>Mot de passe oublié ?</h1>';
    if (!isset($_POST['identifiantMDPOublie']))
    {
        echo '<h5>Veuillez saisir votre identifiant :</h5>
    </div>

    <div class="auth-contenu-form">
        <form method="POST" action="index.php?uc=authentification&action=MDPOublie">
            <table class="mdpoublie-table">
                <tr>
                    <td><input name="identifiantMDPOublie" type="text" placeholder="Saisissez
identifiant ou votre adresse e-mail" id="identifiantMDPOublie" size="37" maxlength="40"></td>
                </tr>
                <tr>
                    <td><button type="submit" class="btn btn-primary
valider">Valider</button></td>
                </tr>
            </table>
        </form>
    </div>

```

```

        </table>

        </form>

    </div>

    <div class="auth-contenu-retour-mdpoublie">

        <table>

            <tr>

                <td><a href="index.php?uc=authentification">Retour</a></td>

            </tr>

        </table>

    </div>;

}

else

{

    if ($idMDPOublieVerifValid == "Administrateur")

    {

        if (!isset($_POST['codeVerifMDPOublieSaisi']))

        {

            echo '<div class="mdpoublie-reussi"><h5>Identifiant valide, veuillez saisir le

code de vérification reçu :</h5></div>

            </div>

            <div class="verifemail-form-mdpoublie">

                <form method="POST"

action="index.php?uc=authentification&action=MDPOublie">

                    <table class="mdpoublie-table">

                        <tr>

                            <td><input name="codeVerifMDPOublieSaisi" type="text"

placeholder="Saisissez le code de vérification reçu" id="codeVerifMDPOublieSaisi" size="37" maxlength="40"><input type="hidden"

name="identifiantMDPOublie" id="identifiantMDPOublie" value="'. $EmailMDPOublie. '"><input type="hidden" name="codeVerifMDPOublie"

id="codeVerifMDPOublie" value="'. $codeVerifEmailMDPOublie. '"><input type="hidden" name="DroitMDPOublie" id="DroitMDPOublie"

value="'. $idMDPOublieVerifValid. '"></td>

                            </tr>

                            <tr>

                                <td><button type="submit" class="btn btn-primary

valider">Valider</button></td>

                                </tr>

                            </table>

                        </form>

                    </div>

                    <div class="auth-contenu-retour-mdpoublie">

                        <table>

                            <tr>

                                <td><a href="index.php?uc=authentification">Retour</a></td>

                                </tr>

                            </table>

                        </div>;

                    }

                }

            }

        }

    }

}

```



```

else
{
    if ($validCodeMDPOublie == false)
    {
        echo '<div class="mdpoublie-echoue"><h5>Vérification échouée, Veuillez
resaisir le code de vérification reçu :</h5></div>

</div>

<div class="verifemail-form-mdpoublie">
    <form method="POST"
action="index.php?uc=authentification&action=MDPOublie">

        <table class="mdpoublie-table">
            <tr>
                <td><input name="codeVerifMDPOublieSaisi" type="text"
placeholder="Saisissez le code de vérification reçu" id="codeVerifMDPOublieSaisi" size="37" maxlength="40"><input type="hidden"
name="identifiantMDPOublie" id="identifiantMDPOublie" value="'. $EmailMDPOublie.'"><input type="hidden" name="codeVerifMDPOublie"
id="codeVerifMDPOublie" value="'. $codeVerifEmailMDPOublie.'"><input type="hidden" name="DroitMDPOublie" id="DroitMDPOublie"
value="'. $idMDPOublieVerifValid.'"></td>

                </tr>
                <tr>
                    <td><button type="submit" class="btn btn-primary
valider">Valider</button></td>

                </tr>
            </table>
        </form>
    </div>

    <div class="auth-contenu-retour-mdpoublie">
        <table>
            <tr>
                <td><a href="index.php?uc=authentification">Retour</a></td>
            </tr>
        </table>
    </div>';
    }
    else
    {
        if (!isset($_POST['changerMDP']) && !isset($_POST['confirmerChangerMDP']))
        {
            echo '<div class="mdpoublie-reussi"><h5>Vérification réussi, Veuillez
saisir votre nouveau mot de passe :</h5></div>

</div>

<div class="verifemail-form-mdpoublie-changermdp">
    <form method="POST"
action="index.php?uc=authentification&action=MDPOublie">

        <table class="mdpoublie-table">
            <tr>
                <th>Mot de passe :</th>

```

```

                <td><input name="changerMDP" type="password"
placeholder="Saisissez votre mot de passe" id="changerMDP" size="23" maxlength="25"></td>

            </tr>

            <tr>

                <th>Confirmer mot de passe :</th>

                <td><input name="confirmerChangerMDP" type="password"
placeholder="Resaisissez votre mot de passe" id="confirmerChangerMDP" size="25" maxlength="25"></td>

            </tr>

            <tr>

                <td colspan="2"><button type="submit" class="btn btn-
primary validerchangermdp">Valider</button><input type="hidden" name="identifiantMDPOublie" id="identifiantMDPOublie"
value="'. $idMDPOublieVerifValid.'"><input type="hidden" name="codeVerifMDPOublieSaisi" id="codeVerifMDPOublieSaisi"
value="'. $codeVerifSaisiEmailMDPOublie.'"><input type="hidden" name="validCodeMDPOublie" id="validCodeMDPOublie"
value="'. $validCodeMDPOublie.'"><input type="hidden" name="emailMDPOublie" id="emailMDPOublie" value="'. $EmailMDPOublie.'"></td>

            </tr>

        </table>

    </form>

</div>

<div class="auth-contenu-retour-changermdp">

    <table>

        <tr>

            <td><a href="index.php?uc=authentification">Retour</a></td>

        </tr>

    </table>

</div>;

}

else

{

    if ($etatChangementMDP == true)

    {

        echo '<div class="mdpoublie-reussi"><h5>Changement de mot de passe
réussi, vous pouvez maintenant vous connecter avec votre nouveau mot de passe.</h5></div>

        </div>

        <form method="POST" action="index.php?uc=authentification"><button
type="submit" class="btn btn-primary ok">Ok</button></form>;

    }

    else

    {

        echo '<div class="mdpoublie-echoue-long"><h5>Changement de mot de
passe échoué, veuillez vérifier que votre mot de passe : <br></br> - Ne comporte pas des caractères interdits ("[]()~!#{};|<=>+"), ni
d\'espaces. </br></br> - Qu\'il dispose d\'une taille valide (entre 8 caractères minimum et 25 maximum). </br></br> - Contient au moins
une majuscule et un chiffre. </br></br> - Qu\'il est bien le même dans le premier champ et dans le second champ de confirmation.
</br></br> - Contient au moins une majuscule et un chiffre. </br></br> - Ne correspond pas au format e-mail
("nom@domaine.nomdedomaine").</h5></div>

        </div>

```

```

        <form method="POST"
action="index.php?uc=authentification&action=MDPOublie"><button type="submit" class="btn btn-primary ok">Ok</button><input type="hidden"
name="identifiantMDPOublie" id="identifiantMDPOublie" value="'. $idMDPOublieVerifValid.'"><input type="hidden"
name="codeVerifMDPOublieSaisi" id="codeVerifMDPOublieSaisi" value="'. $codeVerifSaisiEmailMDPOublie.'"><input type="hidden"
name="validCodeMDPOublie" id="validCodeMDPOublie" value="'. $validCodeMDPOublie.'"><input type="hidden" name="emailMDPOublie"
id="emailMDPOublie" value="'. $EmailMDPOublie.'"></form>';

        }

    }

}

}

else if ($idMDPOublieVerifValid == "Élève" || $idMDPOublieVerifValid == "Professeur")
{
    echo '<div class="mdpoublie-reussi"><h5>Votre identifiant est valide mais le droit
accordé à votre compte ('. $idMDPOublieVerifValid.') ne vous permet pas de changer votre mot de passe. Un mail a été envoyé à
l\'administrateur pour qu\'il change votre mot de passe et vous le retransmettre.</h5></div>

    </div>

    <form method="POST" action="index.php?uc=authentification"><button type="submit"
class="btn btn-primary ok">Ok</button></form>';

    }

    else

    {

        echo '<div class="mdpoublie-echoue"><h5>Identifiant invalide, veuillez vérifier et
ressaisir votre identifiant.</h5></div>

        </div>

        <form method="POST"
action="index.php?uc=authentification&action=MDPOublie"><button type="submit" class="btn btn-primary ok">Ok</button></form>';

        }

    }

}

if (isset($_REQUEST['action']) && $_REQUEST['action'] == "ValidierUtil")
{
    if ($cptEtatValid == 0)
    {
        if (!isset($_POST['authChangerEmailMDP']) && !isset($_POST['erreurRenvoiOriginalEmail']) &&
!isset($_POST['erreurRenvoiOriginalCodeVerif']))
        {
            echo '<div class="auth-contenu-titres-validationechoue">

                <h1>Validation :</h1>

                <h5>Validation échouée, veuillez vérifier que votre adresse e-mail a bien été saisie dans
le premier et dans le second champ de confirmation de façon identique.</h5>

            </div>

            <form method="POST" action="index.php?uc=authentification"><button type="submit" class="btn btn-
primary ok">Ok</button></form>';

        }
    }
}

```

```

else
{
    echo '<div class="auth-contenu-titres-validationechoue">

        <h1>Validation :</h1>

        <h5>Validation échouée, veuillez vérifier que votre adresse e-mail a bien été saisie dans
le premier et dans le second champ de confirmation de façon identique.</h5>

    </div>

    <form method="POST" action="index.php?uc=authentification"><button type="submit" class="btn btn-
primary ok">Ok</button><input type="hidden" name="changerEmailMDPValid" id="changerEmailMDPValid" value="'. $validMDP.'"><input
type="hidden" name="emailOriginal" id="emailOriginal" value="'. $erreurValidEmail.'"><input type="hidden" name="codeVerifOriginal"
id="codeVerifOriginal" value="'. $erreurValidCodeVerif.'">; if (isset($_POST['retourConfirmationErreur'])) { echo '<input type="hidden"
name="changerEmailValid" id="changerEmailValid" value="'. $_POST['retourConfirmationErreur'].'">; } if
(isset($_POST['retourConfirmationCodeSaisiEchoueErreur'])) { echo '<input type="hidden" name="codeVerifSaisiEchoue"
id="codeVerifSaisiEchoue" value="'. $_POST['retourConfirmationCodeSaisiEchoueErreur'].'">; } echo '</form>';

    }

}

if ($cptEtatValid == 1)
{
    echo '<div class="auth-contenu-titres-validationechoue">

        <h1>Validation :</h1>

        <h5>Validation échouée, veuillez vérifier que votre mot de passe a bien été saisie dans le
premier et dans le second champ de confirmation de façon identique.</h5>

    </div>

    <form method="POST" action="index.php?uc=authentification"><button type="submit" class="btn btn-
primary ok">Ok</button></form>;

    }

if ($cptEtatValid == 2)
{
    echo '<div class="auth-contenu-titres-validationechoue">

        <h1>Validation :</h1>

        <h5>Validation échouée, veuillez vérifier que votre mot de passe ne correspond pas à votre
adresse e-mail.</h5>

    </div>

    <form method="POST" action="index.php?uc=authentification"><button type="submit" class="btn btn-
primary ok">Ok</button></form>;

    }

if ($cptEtatValid == 3)
{
    if (!isset($_POST['authChangerEmailMDP']) && !isset($_POST['erreurRenvoiOriginalEmail']) &&
!isset($_POST['erreurRenvoiOriginalCodeVerif']))

    {
        echo '<div class="auth-contenu-titres-validationechoue">

            <h1>Validation :</h1>

```

```
<h5>Validation échouée, veuillez vérifier que votre adresse e-mail : </br></br> - Ne
comporte pas des caractères interdits ("/[>()~!#{};|<=>+") , ni de majuscules, ni d'espaces. </br></br> - Qu'elle dispose d'un format
("nom@domaine.nomdedomaine") et d'une taille (entre 10 caractères minimum et 40 maximum) valides.</h5>
```

```
</div>
```

```
<form method="POST" action="index.php?uc=authentification"><button type="submit"
class="btn btn-primary ok">Ok</button></form>;
```

```
}
```

```
else
```

```
{
```

```
    echo '<div class="auth-contenu-titres-validationechoue">
```

```
        <h1>Validation :</h1>
```

```
        <h5>Validation échouée, veuillez vérifier que votre adresse e-mail : </br></br> - Ne
comporte pas des caractères interdits ("/[>()~!#{};|<=>+") , ni de majuscules, ni d'espaces. </br></br> - Qu'elle dispose d'un format
("nom@domaine.nomdedomaine") et d'une taille (entre 10 caractères minimum et 40 maximum) valides.</h5>
```

```
    </div>
```

```
    <form method="POST" action="index.php?uc=authentification"><button type="submit" class="btn btn-
primary ok">Ok</button><input type="hidden" name="changerEmailMDPValid" id="changerEmailMDPValid" value="'. $validMDP.'"><input
type="hidden" name="emailOriginal" id="emailOriginal" value="'. $erreurValidEmail.'"><input type="hidden" name="codeVerifOriginal"
id="codeVerifOriginal" value="'. $erreurValidCodeVerif.'">; if (isset($_POST['retourConfirmationErreur'])) { echo '<input type="hidden"
name="changerEmailValid" id="changerEmailValid" value="'. $_POST['retourConfirmationErreur'].'">; } if
(isset($_POST['retourConfirmationCodeSaisiEchoueErreur'])) { echo '<input type="hidden" name="codeVerifSaisiEchoue"
id="codeVerifSaisiEchoue" value="'. $_POST['retourConfirmationCodeSaisiEchoueErreur'].'">; } echo '</form>;
```

```
}
```

```
}
```

```
if ($cptEtatValid == 4)
```

```
{
```

```
    echo '<div class="auth-contenu-titres-validationechoue">
```

```
        <h1>Validation :</h1>
```

```
        <h5>Validation échouée, veuillez vérifier que votre mot de passe : </br></br> - Ne comporte
pas des caractères interdits ("/[>()~!#{};|<=>+") , ni d'espaces. </br></br> - Qu'il dispose d'une taille valide (entre 8 caractères
minimum et 25 maximum). </br></br> - Contient au moins une majuscule et un chiffre. </br></br> - Ne correspond pas au format e-mail
("nom@domaine.nomdedomaine").</h5>
```

```
    </div>
```

```
    <form method="POST" action="index.php?uc=authentification"><button type="submit" class="btn btn-
primary ok">Ok</button></form>;
```

```
}
```

```
if ($cptEtatValid == 5)
```

```
{
```

```
    echo '<div class="auth-contenu-titres-validationreussi">;
```

```
        if (!isset($_POST['authChangerEmailMDP']) && !isset($_POST['retourConfirmation']) &&
!isset($_POST['codeVerifEmailSaisi']) && !isset($_POST['retourConfirmationCodeSaisiEchoue']))
```

```
    {
```

```
        echo '<h1>Validation :</h1>
```

```

        <h5>Validation réussie, veuillez vérifier votre adresse e-mail en saisissant le code de
vérification reçu :</h5>';

    }

    else
    {
        if (isset($_POST['authChangerEmailMDP']) || isset($_POST['retourConfirmation']))
        {
            echo ' <h1>Changer adresse e-mail :</h1>

            <h5>Changement d\'adresse e-mail réussi, veuillez vérifier la nouvelle adresse e-mail en
saisissant le code de vérification reçu :</h5>';

        }

        if (isset($_POST['codeVerifEmailSaisi']) && $verifCodeEgal == false ||
isset($_POST['retourConfirmationCodeSaisiEchoue']) && $verifCodeEgal == false)
        {
            echo ' <h1>Vérification E-mail :</h1>

            <div class="auth-contenu-titres-validationechouetitre"><h5>La vérification par code de
l\'adresse e-mail saisie a échoué, veuillez ressaisir le code de vérification reçu :</h5></div>';

        }

        if ($verifCodeEgal == true)
        {
            echo ' <h1>Vérification E-mail Réussi :</h1>

            <h5>La vérification par code de l\'adresse e-mail saisie a réussi, vous pouvez maintenant
vous connecter à votre compte en tant qu\'administrateur.</h5>';

        }

    }

    if ($verifCodeEgal == false)
    {
        echo ' </div>

        <div class="verifemail-form">

            <form method="POST" action="index.php?uc=authentification&action=ValidierUtil">

                <input name="codeVerifEmailSaisi" type="text" placeholder="Saisissez le code de
vérification reçu" id="codeVerifEmailSaisi" size="30">

                <input type="hidden" name="validMDPSaisi" id="validMDPSaisi" value="'. $validMDP.'">

                <input type="hidden" name="validEmailSaisi" id="validEmailSaisi"

value="'. $validEmail.'">

                <input type="hidden" name="validCodeVerifSaisi" id="validCodeVerifSaisi"

value="'. $codeVerifEmail.'">

                <button type="submit" class="btn btn-primary ok">Valider</button>

            </form>

        </div>

        <div class="verifemail-changeremail">

            <form method="POST" action="index.php?uc=authentification">

                <p>Code de vérification non reçu ?</p>

```

```

        <input type="hidden" name="changerEmailMDPValid" id="changerEmailMDPValid"
value="'. $validMDP.'">

        <input type="hidden" name="emailOriginal" id="emailOriginal" value="'. $validEmail.'">
        <input type="hidden" name="codeVerifOriginal" id="codeVerifOriginal"
value="'. $codeVerifEmail.'">;

        if (isset($_POST['authChangerEmailMDP']) || isset($_POST['retourConfirmation']))
        {
            echo '<input type="hidden" name="changerEmailValid" id="changerEmailValid"
value="1">';

        }

        if (isset($_POST['codeVerifEmailSaisi']) ||
isset($_POST['retourConfirmationCodeSaisiEchoue']))
        {
            echo '<input type="hidden" name="codeVerifSaisiEchoue" id="codeVerifSaisiEchoue"
value="1">';

        }

        echo '<button type="submit" class="btn btn-primary ok">Changer mon adresse e-
mail.</button>

        </form>
    </div>

    <div class="auth-contenu-retour-valid">
        <table>
            <tr>
                <td><a href="index.php?uc=authentification">Retour</a></td>
            </tr>
        </table>
    </div>';
}
else
{
    echo '</div>

    <form method="POST" action="index.php?uc=accueil"><button type="submit" class="btn btn-
primary ok">Ok</button></form>';

}

    }

    }

    ?>
</div>
</div>
</div>

<div class="container auth-finpage" draggable="false" ondragstart="return false;">
    <p class="finpage-texte">2022 © Sainte Agnès - Jeanne d'Arc (ESC)</p>

```

```
    </div>

</div>

<div class="container-fluid auth-corps-bg-aide" draggable="false" ondragstart="return false;"></div>
</body>
</html>
```