

Portfolio

Laura HUBER

Table des matières

Table des matières	2
Introduction	3
1 Expérience	4
2 Formation	5
3 Compétences	6
4 Présentation de l'entreprise	7
5 Organigramme	9
6 Le Département des Systèmes d'Information	10
7 Mes missions	11
8 Portefeuille d'activités professionnelles	12

Introduction

Je m'appelle Laura Huber et je suis actuellement apprentie au sein de l'unité fonctionnelle de la direction interrégionale des services pénitentiaires de Paris dans le cadre d'un BTS Services Informatiques aux Organisations option Solutions Logicielles et Applications Métiers (SLAM).

1 Expérience

Direction interrégionale des services pénitentiaires de Paris

Alternante en développement informatique

Avril 2021 - maintenant

- Création d'une page web (annuaire/catalogue des services de la DSI)
- Développement d'une application de bureau (base des agents)

Traduction Français/Anglais/Alemand

Activité indépendante

Septembre 2018 - février 2020 (1,5 ans)

- Entreprenariat
- Gestion de temps

Restaurant Mobilis in Mobili

Employée polyvalente

Novembre 2017 - octobre 2018 (1 an)

- Accueil, renseignement des clients, prise et service des commandes
- Préparation des commandes, découpage des légumes, cuisson, dressage, écaillage
- Approvisionnement, organisation de la présentation des produits
- Étiquetage et rangement des produits
- Entretien de la salle du restaurant, de la cuisine, des toilettes et des équipements, plonge

Supermarché HIT

Employée polyvalente

Novembre - décembre 2015 (2 mois)

- Gestion de la caisse et compétences de base en comptabilité
- Rangement des produits et réassort de la marchandise dans les rayons
- Orienter et renseigner les clients

Hospice Kliniken Wied

Stage

Janvier 2014 (2 semaines)

- Découverte de l'ergothérapie
- Aider les gens à surmonter des dépendances
- Observer et communiquer avec les patients et le personnel

2 Formation

Brevet de technicien supérieur (BTS)

Services informatiques aux organisations option SLAM

2020-2022

Licence 2

LEA Anglais-Suédois

2016-2018

3 Compétences

Développement

Basique :

- C++
- Java
- Python
- MySQL
- PHP
- Qt

Intermédiaire :

- HTML
- CSS
- Bootstrap

Outils

- Environnements Windows et Linux
- Bureautique Microsoft
- Unity
- Unreal Engine 3
- JIRA
- Git

Langues

- Français (courant)
- Anglais (courant)
- Allemand (native)
- Diplôme d'étude du Latin

Centres d'intérêt

- Jeux vidéo et jeux de société
- Game jams
- Suivi occasionnel de MOOC

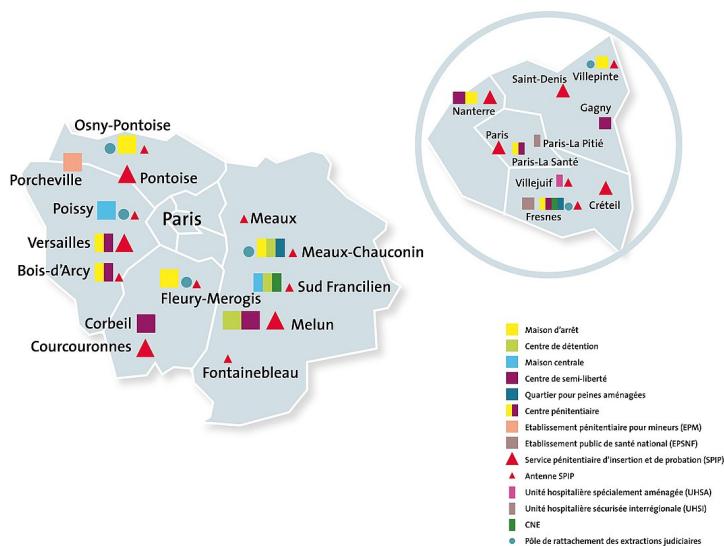
4 Présentation de l'entreprise

La DISP de Paris

La Direction Interrégionale des Services Pénitentiaires de Paris est un service dé-concentré du ministère de la Justice, relevant de la direction de l'administration pénitentiaire.

Elle est l'une des dix directions interrégionales des services pénitentiaires présentes sur le territoire métropolitain et d'outre-mer.

Elle anime, coordonne et contrôle l'activité des établissements et des services pénitentiaires placés sous son autorité, en regroupant les 8 départements franciliens, sur le ressort des cours d'appel de Paris (75) et de Versailles (78).

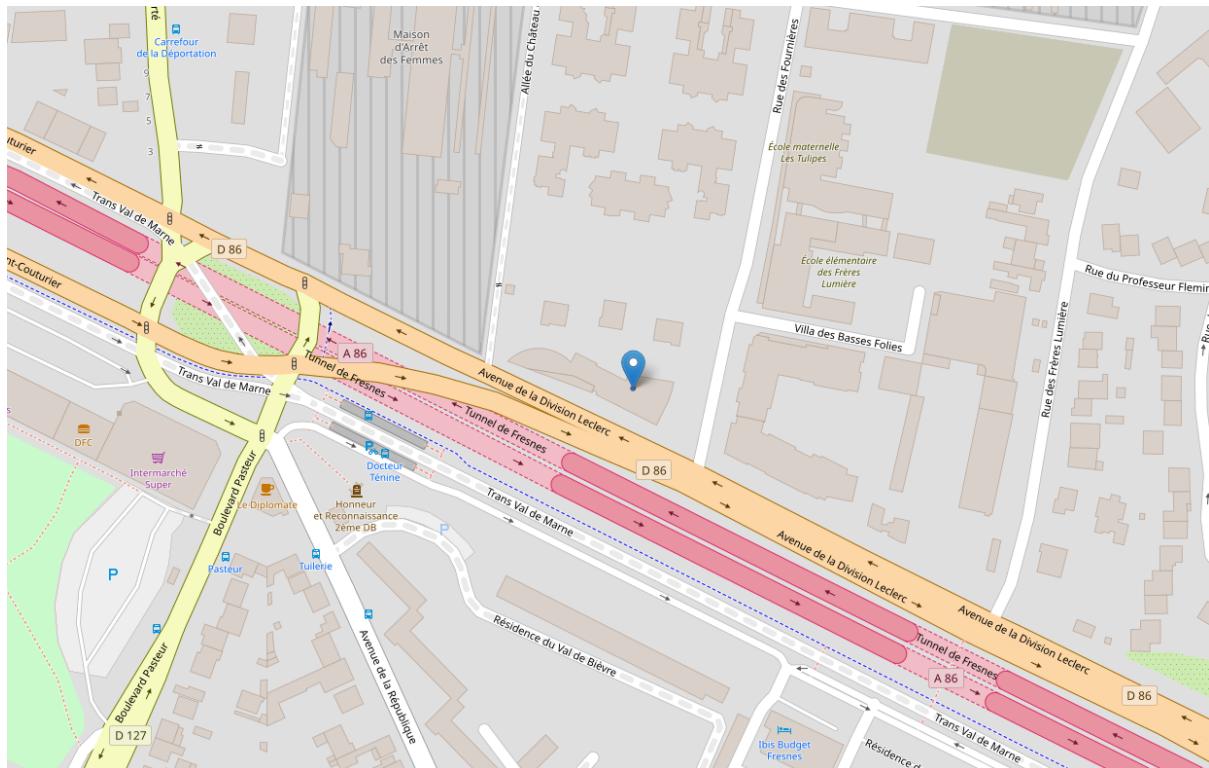


Les enjeux de la DISP de Paris sont :

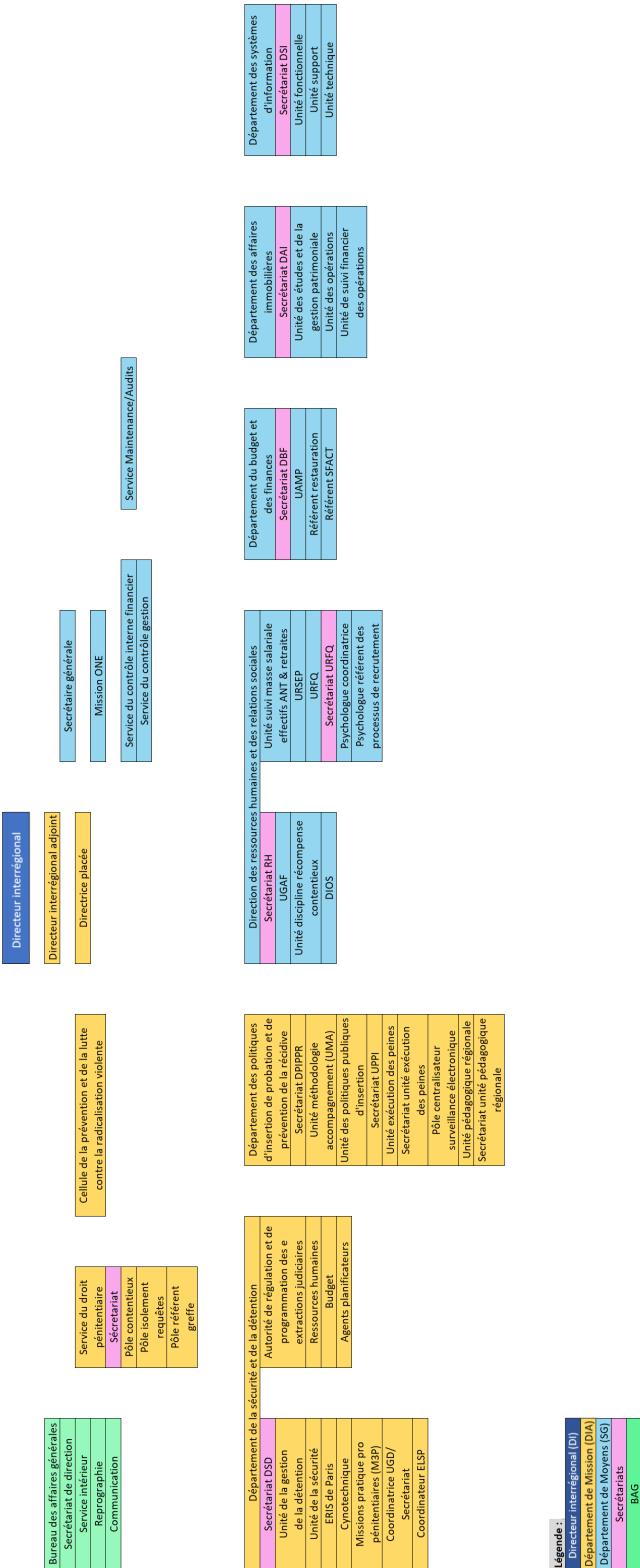
- La stabilisation de la population carcérale
- La structuration de la coordination santé justice
- La lutte contre la radicalisation en milieu fermé et ouvert
- Le développement des aménagements de peine
- Le plan interrégional de lutte contre les violences
- La prévention du suicide

Coordonnées

DISP PARIS
3 Avenue de la Division Leclerc
BP 103
94267 Fresnes Cedex
01.88.28.70.00



5 Organigramme



6 Le Département des Systèmes d'Information

Le département des systèmes d'information (DSI) est chargé de mettre en oeuvre les applications informatiques nationales et d'assurer la cohérence des systèmes d'informations interrégionaux.

Le département applique toutes les procédures et dispositifs pour assurer la sécurité des systèmes d'information (SSI).

Le DSI détermine, sur la base des orientations définies par l'administration centrale, les modalités de mise en œuvre et de maintenance des infrastructures informatiques, des applications nationales, des activités de support aux utilisateurs et de la sécurité des systèmes d'information. Il associe les autres départements à la conception et à la réalisation des actions interrégionales, ainsi qu'au suivi et à l'évaluation de leur exécution.

En liaison avec les chefs de service et les directeurs des services pénitentiaires d'insertion et de probation, il évalue les besoins en patrimoine informatique, assure la programmation et le financement des nouvelles acquisitions et des renouvellements et supervise leur déploiement.

Il anime le réseau des correspondants locaux du système d'information, situés dans les établissements pénitentiaires et les SPIP, coordonne l'accomplissement de leurs missions et anime leurs actions. Il contribue à la formation des agents aux applications informatiques régionales et apporte conseil et assistance fonctionnelle.

7 Mes missions

Ma première mission consistait en la mise en place d'un annuaire de la DSI sous la forme d'une page web dans laquelle est intégré un catalogue des services proposés. Le but du site est de rendre la DSI plus visible et de faciliter et automatiser les démarches courantes des utilisateurs à l'aide de la gestion du flux de travaux. Pour cela, je me servais de WAMP et de WordPress ainsi que de HTML, CSS, PHP, TCPDF et MySQL.

Maintenant, je suis responsable du développement d'un logiciel de base des agents. Il s'agissait initialement de la transformation d'une base de données Access en application de bureau. Au cours de la mission, nous avons décidé d'opter plutôt pour une application web. La base de données en question contient les fiches des agents des établissements de Paris et doit être modernisée.

Au sein du CFA, nous avons mis en place un MédiaWiki sur "Les métiers de l'informatique" dans le contexte du PPE 3. Ce projet était à réaliser en groupe.

Dans le contexte de la M2L (Maison des Ligues de Lorraine), j'ai créé un site web, la "bibliothèque Sterling", sur lequel les utilisateurs peuvent télécharger des livres numériques en format PDF.

Dans ce même contexte, j'ai développé une application de bureau nommé "Pixters" qui permet de s'occuper d'un animal de compagnie virtuel.

J'ai également fait une veille technologique sur l'apprentissage profond.

8 Portefeuille d'activités professionnelles



Direction Interrégionale des
Services Pénitentiaires de Paris

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS

SESSION 2022

Tableau de synthèse des réalisations professionnelles

NOM et prénom : HUBER Laura	N° candidat : 02142367763
Centre de formation : Faculté de Métiers de l'Essonne	Option : <input type="checkbox"/> SISR <input checked="" type="checkbox"/> SLAM

Adresse URL du portfolio : <https://lumicco.github.io/Portfolio/>

Compétences mises en œuvre	Période (sous la forme du JJ/MM/AA au JJ/MM/AA)	Gérer le patrimoine informatique	Répondre aux incidents et aux demandes d'assistance et d'évolution	Développer la présence en ligne de l'organisation	Travailler en mode projet	Mettre à disposition des utilisateurs un service informatique	Organiser son développement professionnel
		<ul style="list-style-type: none"> ▪ Recenser et identifier les ressources numériques ▪ Exploiter des référentiels, normes et standards adoptés par le prestataire informatique ▪ Mettre en place et vérifier les niveaux d'habilitation associés à un service ▪ Vérifier les conditions de la continuité d'un service informatique ▪ Gérer des sauvegardes ▪ Vérifier le respect des règles d'utilisation des ressources ▪ Traiter des demandes concernant les services réseau et système, applicatifs ▪ Traiter des demandes concernant les applications ▪ Participer à la valorisation de l'image de l'organisation sur les médias numériques en tenant compte du cadre juridique et des enjeux économiques ▪ Référencer les services en ligne de l'organisation et mesurer leur visibilité. ▪ Participer à l'évolution d'un site Web exploitant les données de l'organisation. ▪ Analyser les objectifs et les modalités d'organisation d'un projet ▪ Planifier les activités ▪ Évaluer les indicateurs de suivi d'un projet et analyser les écarts ▪ Réaliser les tests d'intégration et d'acceptation d'un service ▪ Déployer un service ▪ Accompagner les utilisateurs dans la mise en place d'un service 					
Réalisations professionnelles (intitulé et liste des documents et productions associés)							
Mise en place d'un MédiaWiki (Fiche d'activité n°3, PPE 3)	de 03/21 au 06/21				X	X	X
Développement d'un site web (Fiche d'activité n°4, PPE 4)	de 11/21 au 04/22	X		X			X
Développement d'une application (Fiche d'activité n°5, PPE 5)	de 11/21 au 04/22	X					X
Réalisations en milieu professionnel en cours de première année							
Projet de développement web - site web (Fiche d'activité n°1)	de 05/21 au 12/21			X			X
Réalisations en milieu professionnel en cours de seconde année							
Projet de développement web - application web (Fiche d'activité n°2)	de 12/21 au 06/22	X	X			X	X

Compte rendu d'Activité		Fiche n° 1
Libellé court	Compte rendu mission entreprise	
Description	Compte rendu d'un projet de développement web.	
Contexte	Mission accomplie au sein de l'unité fonctionnelle de la DISP de Paris.	

Localisation	<input checked="" type="checkbox"/> Entreprise <input type="checkbox"/> CFA
Source	<input type="checkbox"/> 1ère année <input checked="" type="checkbox"/> 2ème année <input type="checkbox"/> TP <input type="checkbox"/> PPE
Cadre	<input checked="" type="checkbox"/> Seule <input type="checkbox"/> En équipe
Type	<input checked="" type="checkbox"/> Vécu <input type="checkbox"/> Observé <input type="checkbox"/> Simulé

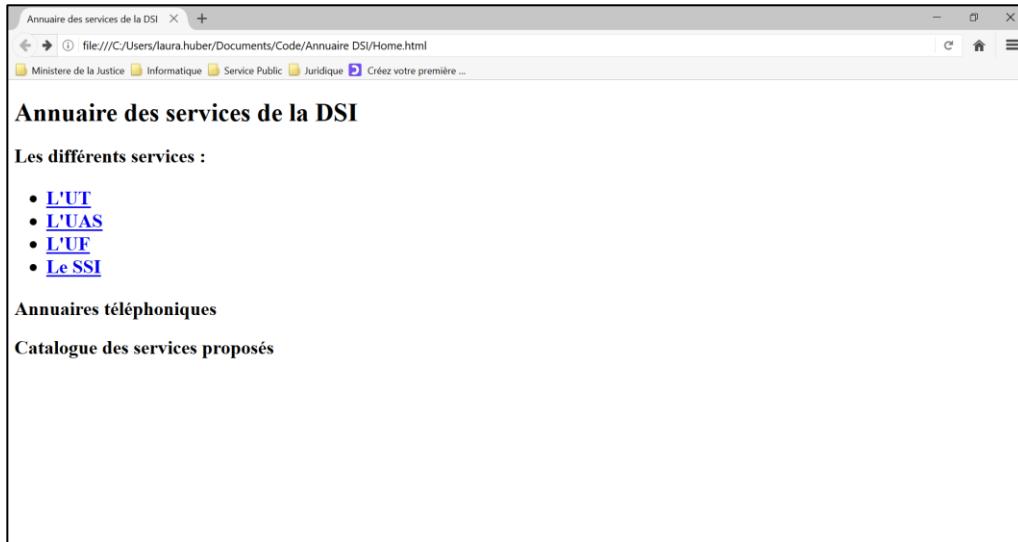
Environnement Technologique	WordPress, WAMP, Test Mail Server Tool, phpMyAdmin
Moyens	HTML5, CSS, PHP7 + TCPDF, MySQL
Avis personnel	Projet web assez restrictif à cause de l'utilisation de WordPress et une charge importante de travail que je trouve trop lourde pour une personne seule. Pas d'aide sur le plan technique, j'ai donc été obligée de tout apprendre moi-même sans connaissances préalables ce qui était très frustrant. Par ailleurs, je n'aime pas du tout le développement web et le fait qu'on m'a donné un projet web même après avoir demandé mes préférences me donne simplement l'impression de ne pas être prise au sérieux.

Activités mises en œuvre pour la réalisation de cette situation	
	Organisation de son développement professionnel
	Développement la présence en ligne de l'organisation
	Conception et développement d'une solution applicative
	Maintenance corrective et évolutive d'une solution applicative
	Gestion des données

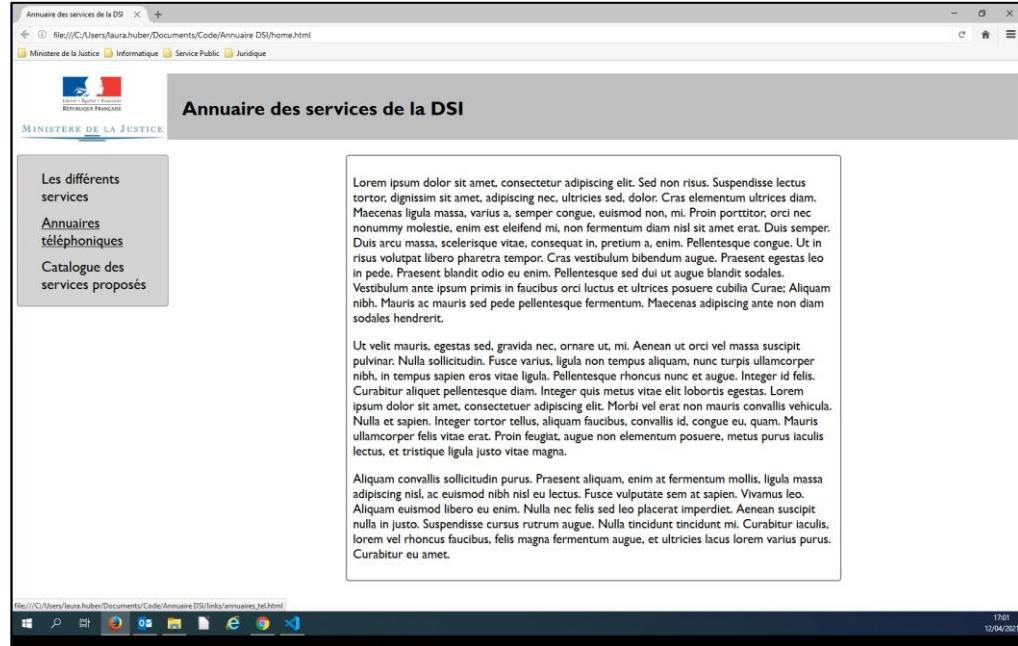
Déroulement de l'activité :

Ma mission consiste en la création d'une application intranet pour un annuaire des services de la DSI de la DISP de PARIS.

Tout d'abord, je réfléchissais à un site intranet qui présenterait la DSI et les services traités par les différentes unités sans remplacer l'intranet déjà existant. Pour cela, j'ai mis en place une maquette simple en apprenant les bases de HTML5 et CSS3.

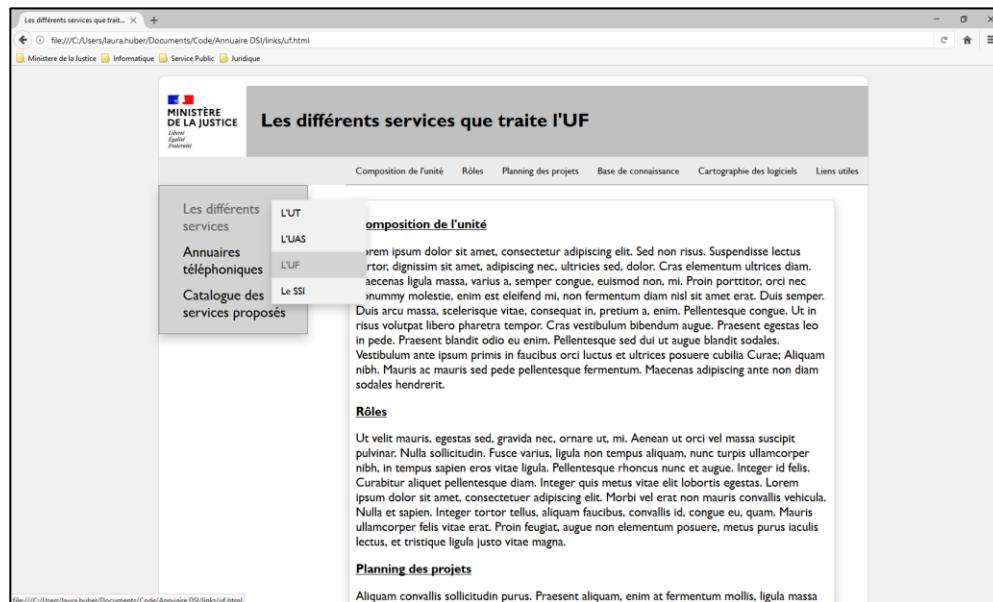


Premiers pas avec HTML



Premiers pas avec CSS et structuration de la page avec la méthode Flexbox (conteneurs)

Finalement, j'ai pu créer un page web rudimentaire qui servait comme squelette pour la suite.



Maquette HTML

```
<!DOCTYPE html>
<html lang="fr">
    <head>
        <meta charset="utf-8" />
        <link rel="stylesheet" href="style.css" />
        <title>Annuaire des services de la DSI</title>
    </head>

    <body>
        <!-- HEADER -->
        <header>
            <p>
                <a href="index.html"></a>
            </p>
            <h1><br /> &nbsp; Annuaire des services de la DSI</h1>
        </header>

        <div class="container">
            <!-- NAVIGATION BAR (SIDE) -->
            <nav id="sidenav">
                <ul>
                    <div class="dropdown">
                        <li class="dropbutton">Les différents services</li>
                        <!-- DROPDOWN MENU -->
                        <div class="dropdown-content">
                            <li><a href="links/ut.html">L'UT </a></li>
                            <li><a href="links/uas.html">L'UAS </a></li>
                            <li><a href="links/uf.html">L'UF </a></li>
                            <li><a href="links/ssi.html">Le SSI </a></li>
                        </div>
                    </div>
                    <li><a href="links/annuaires_tel.html">Annuaires téléphoniques</a></li>
                    <li><a href="links/catalogue.html">Catalogue des services proposés</a></li>
                </ul>
            </nav>

            <!-- CONTENT -->
            <section id="content">
                <p>Contenu</p>
            </section>
        </div>
    </body>
</html>
```

Le HTML de la maquette

```

html
{
    font-family:'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;
    background-color: #fififi;
}

/*LOGO SIZE AND POSITION*/
.logo
{
    height: 120px;
    width: auto;
    float: left;
}

/*HEADER STYLE*/
header
{
    font-size: 1em;
    background-color: silver;
    min-height: 120px;
    margin-top: 0;
    /* margin-left: 15px; */
}

/*REDUCE WHITE BAR OVER HEADER*/
header h1
{
    margin-top: 0;
}

/*BODY STYLE AND SIZE*/
body
{
    margin: 1% auto 1% auto;
    width: 50%;
    min-width: 900px;
    padding-right: 15px;
    border: 1px lightgrey solid;
    border-radius: 10px;
    background-color: white;
}

/*SIDEBAR STYLE*/
#sidenav
{
    border: 0.5px grey solid;
    background-color: lightgray;
    width: 180px;
    height: 260px;
    margin-right: 15px;
    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
    /*STICKY*/
    position: -webkit-sticky;
    position: sticky;
    top: 0;
    z-index: 99999;
}

/*TOPBAR STYLE*/
#topnav
{
    clear: both;
    background-color: #ececce;
    display: flex;
    flex-direction: row;
    justify-content: right;
    /*STICKY*/
    position: -webkit-sticky;
    position: sticky;
    top: 0;
    z-index: 99998;
}

/*TOPBAR LINK STYLE*/
#topnav a
{
    font-size: 0.8em;
    text-align: center;
    padding: 14px 16px;
    border-bottom: 1.5px grey solid;
}

/*TOPBAR LINK STYLE ON HOVER*/
#topnav a:hover
{
    background-color: lightgrey;
}

/*DROPDOWN MENU DISPLAY*/
.dropdown
{
    position: relative;
    display: inline-block;
}

/*DROPDOWN MENU STYLE*/
.dropdown-content
{
    display: none;
    position: absolute;
    background-color: #fififi;
    min-width: 160px;
    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
    z-index: 1;
    font-size: 0.7em;
}

```

```

/*DROPDOWN MENU LINK STYLE*/
.dropdown-content a
{
    color: black;
    padding: 12px 16px;
    text-decoration: none;
    display: block;
}

/*DROPDOWN MENU BULLET POINT MARGIN*/
.dropdown-content li
{
    margin: 0;
}

/*MENU DROPODOWN POSITION RIGHT*/
.dropdown:hover .dropdown-content
{
    display: block;
    top: 0;
    left: 100px;
}

/*DROPDOWN MENU BUTTON COLOUR ON HOVER*/
.dropdown:hover .dropbutton
{
    color: rgb(100, 100, 100);
}

/*DROPDOWN MENU LINK COLOUR ON HOVER*/
.dropdown-content a:hover
{
    background-color: lightgrey;
    text-decoration: none;
}

/*LV2 HEADING SIZE*/
h2
{
    font-size: 1em;
    font-weight: bold;
    text-decoration: underline;
}

/*BULLET LIST STYLE*/
ul
{
    font-size: 1em;
    list-style-type: none;
    padding: 0 15px 15px 15px ;
}

/*BULLET POINT STYLE*/
li
{
    margin-bottom: 15px;
}

/*FLEX CONTAINER*/
.container
{
    clear: both;
    display: flex;
    flex-direction: row;
    margin-top: 0;
    padding: 0px 0 30px 0;
}

/*MAIN SECTION STYLE*/
#content
{
    border: 1px lightgrey solid;
    border-radius: 5px;
    min-width: 500px;
    max-width: 800px;
    padding: 10px;
    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
    margin-left: auto;
    margin-right: auto;
    margin-top: 25px;
    font-size: 1.1em;
}

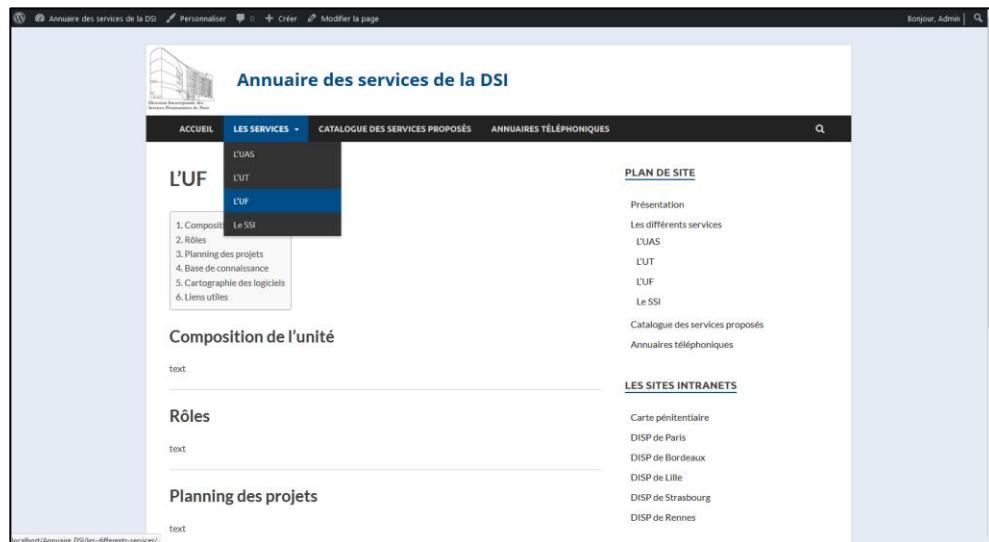
/*LINK STYLE*/
a
{
    color: black;
    text-decoration: none;
}

/*LINK STYLE ON HOVER*/
a:hover
{
    text-decoration: underline;
    color: rgb(100, 100, 100);
}

```

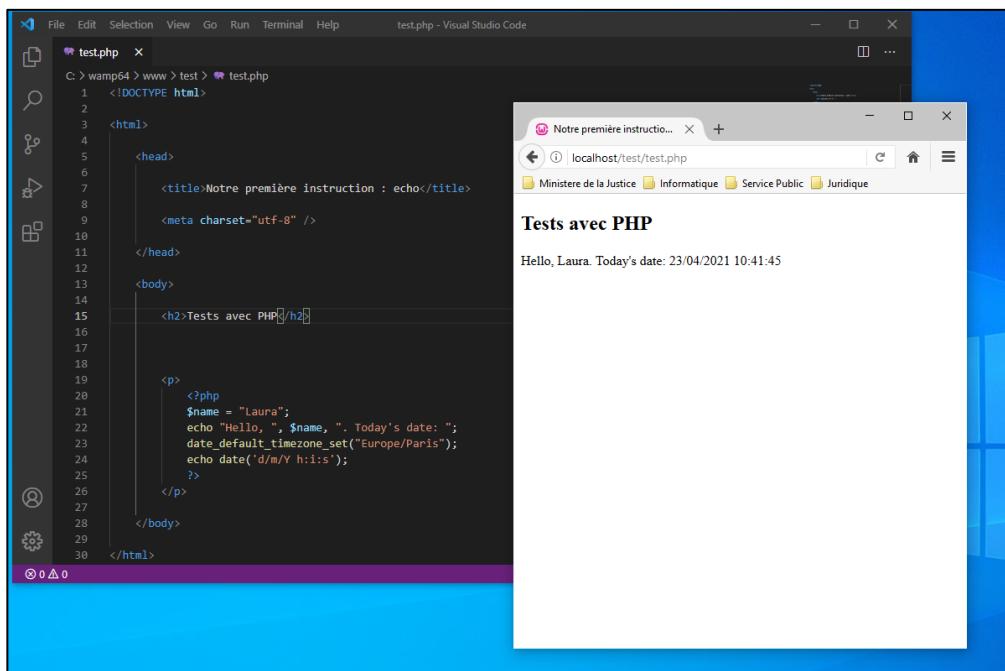
Le CSS de la maquette

Afin de créer une page qui peut être modifiée par les utilisateurs sans passer par le code, il me semblait pratiquement indispensable d'utiliser un CMS comme WordPress et comme il faut entre autres un serveur web, une base de données et un hébergeur, nous avons décidé d'utiliser un serveur WAMP. Après l'installation et la configuration du serveur, j'y ai installé WordPress. De ce fait, j'ai procédé à la mise en place du squelette de la page et de la configuration du CSS. J'ai également ajouté un plan de site et des liens vers les intranets.



Squelette de la page

De façon à connaître les besoins des différents services, j'ai pris rendez-vous avec les chefs d'unité pour leur présenter mon projet et remplir le catalogue de services qu'ils souhaitent intégrer dans l'annuaire. Après cela, j'ai commencé à apprendre les bases de PHP et de MySQL en vue de réaliser les outils qui seront ajoutés au site intranet.



Premiers pas avec PHP

Le premier élément que j'ai mis en place est une page de demande d'habilitation aux applications métiers. A ce titre, j'ai créé un formulaire à remplir en ligne qui, après envoi, envoie un mail au supérieur de l'utilisateur afin qu'il valide ou refuse la demande. En cas de refus, l'utilisateur en est informé par mail et en cas de validation, un mail est envoyé à l'unité fonctionnelle pour habiliter l'utilisateur aux applications demandées. Les informations et l'état d'avancement de la demande sont enregistrés dans une base de données SQL.

Maquette du formulaire de demande d'habilitation

Etat final du formulaire de demande d'habilitation

A screenshot of the MySQL Workbench interface. It shows a table named 'formulaire_agir' with the following data:

	id	demande	date_demande	date_ouverture	nom_res	prenom_res	email_res	tel_res	fonction_res	etab_res	nom_user	prenom_user	email_user	tel_user	habilitations	nom_agent	infos_comp	statut
	1	suppression	02/06/2021	03/06/2021	Machin	Thib	machin@tutu.fr			Machin2	Thib2			machin2@tutu.fr		CAT_CD_DM CAT_FOUL_DQ CAT_DE_DM CAT_OG_DQ C.		Refusé

La base de données associé au formulaire

```
//Connect to database
try
{
    $bdd = new PDO('mysql:host=localhost;dbname=php;charset=utf8', 'root', '');
    array(PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION);
}
catch (Exception $e)
{
    die('Erreur : ' . $e->getMessage());
}

//Prepare query
$req = $bdd->prepare('INSERT INTO formulaire_agir(demande, date_demande, date_ouverture,
nom_res, prenom_res, email_res, tel_user,
nom_user, prenom_user, email_user, tel_user,
habilitations, nom_agent, infos_comp) VALUES(:demande, :date_demande, :date_ouverture,
:nom_res, :prenom_res, :email_res, :tel_res, :fonction_res, :etab_res,
:nom_user, :prenom_user, :email_user, :tel_user,
:habilitations, :nom_agent, :infos_comp)');
$demande = $_POST['demande'];
$date_demande = $_POST['date-demande'];
$date_ouverture = $_POST['date-ouverture'];

$nom_res = $_POST['nom-res'];
$prenom_res = $_POST['prenom-res'];
$email_res = $_POST['email-res'];
$tel_res = $_POST['tel-res'];
$fonction_res = $_POST['fonction-res'];
$etab_res = $_POST['etab-res'];

$nom_user = $_POST['nom-user'];
$prenom_user = $_POST['prenom-user'];
$email_user = $_POST['email-user'];
$tel_user = $_POST['tel-user'];

//Show checked profiles
$habilitations = 'Aucun';
if(isset($_POST['profil'])) && is_array($_POST['profil']) && count($_POST['profil']) > 0)
{
    $habilitations = implode("\r\n", $_POST['profil']);
}

$nom_agent = $_POST['nom-agent'];

//Limit max chars per line (default: 75)
$infos_comp = wordwrap($_POST['infos-comp']);

//Execute query, add data to database
$req->execute(array(
    'demande' => $demande,
    'date_demande' => $date_demande,
    'date_ouverture' => $date_ouverture,
    'nom_res' => $nom_res,
    'prenom_res' => $prenom_res,
    'email_res' => $email_res,
    'tel_res' => $tel_res,
    'fonction_res' => $fonction_res,
    'etab_res' => $etab_res,
    'nom_user' => $nom_user,
    'prenom_user' => $prenom_user,
    'email_user' => $email_user,
    'tel_user' => $tel_user,
    'habilitations' => $habilitations,
    'nom_agent' => $nom_agent,
    'infos_comp' => $infos_comp,
));
}

//Get last ID
$response = $bdd->query('SELECT id FROM formulaire_agir ORDER BY id DESC LIMIT 1');
$array = $response->fetch(PDO::FETCH_ASSOC);

//Convert array to string
$id = implode($array);

//Accept/decline links, send data via get
$accepter = "localhost/Annuaire_DS1/php/Habilitations/AGIR/accepter.php?email_res=$email_res
&email_user=$email_user&id=$id";
$refuser = "localhost/Annuaire_DS1/php/Habilitations/AGIR/refuser.php?email_res=$email_res&id=
$id";
```

Le code PHP qui lie le formulaire et la base de données

Par la suite, le responsable de la reprographie a exprimé le besoin d'un formulaire de commande de supports de communication pour le projet "Numérique en détention" (NED). J'ai donc intégré le catalogue du NED au site et préparé le formulaire demandé.

Supports de communication

Vous pouvez ici commander les supports de communication pour le NED du Portail Famille, cet outil permet aux familles des détenus, de réserver les parloirs sur internet.

Il y a quatre supports de communication :

1. **Affiches pour les détenus** (?)
À placer dans la détention sur les panneaux d'affichage de tous les lieux de rassemblement (Salle de musculation / Salle de culte / Bibliothèque ...)
2. **Affiches pour les familles** (?)
À placer à l'abri famille proche des bornes parloirs, sur les vitres vers l'extérieur et dans les salles d'attente....
3. **Affiches didacticiels (pas à pas)** (?)
À placer au niveau des bornes parloirs et à côté de l'affiche pour les familles.
4. **Les flyers pour les familles** (?)
À distribuer aux familles avec les permis de visites, à mettre à disposition sur des présentoirs.

Pour passer commande, nous vous invitons à cliquer sur le lien suivant : [Commander des affiches](#)

L'onglet du NED sur le site

**MINISTÈRE
DE LA JUSTICE**
*Justice
Égalité
Intégrité*

Formulaire de commande d'affiches

Nom : _____ Prénom : _____

Adresse mail : _____ Établissement : _____

Combien d'affiches souhaitez-vous commander ?

Affiches pour les détenus (40x60) : <input type="text" value="0"/>	Affiches didacticiels (60x80) : <input type="text" value="0"/>
Affiches pour les familles (40x60) : <input type="text" value="0"/>	Flyers (A4 plié en 3) : <input type="text" value="0"/>

ENVOYER © 2021 DISP de Paris/DSI

Le formulaire auquel on peut accéder via l'onglet du NED, l'icône à côté de la taille permet de prévisualiser l'affiche

```
<!DOCTYPE html>
<html lang="fr">
    <head>
        <meta charset="utf-8" />
        <title>Formulaire de commande d'affiches</title>
        <link rel="stylesheet" href="style.css" />
        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
    </head>

    <body>
        

        <form action="action_page.php" method="post" name="form">
            <h1>Formulaire de commande d'affiches</h1>

            <div class="align">
                <label>Nom :</label>
                <input type="text" id="nom" name="nom" required>
            </div>

            <div>
                <label>Prénom :</label>
                <input type="text" id="prenom" name="prenom" required>
            </div>

            <div class="align">
                <label>Adresse mail :</label>
                <input type="email" id="email" name="email" required>
            </div>

            <div>
                <label>Établissement :</label>
                <input type="text" id="etablissement" name="etablissement" required>
            </div>

            <p><br><b>Combien d'affiches souhaitez-vous commander ?</b></p>

            <div class="align">
                <label>Affiches pour les détenus
                    <a title="Aperçu" href="http://localhost/Annuaire_DSI/wp-content/uploads/2021/08/affiche_detenu.png" target="_blank" rel="noopener"><span>(40x60)</span>
                        <i style="font-size:12px; color:grey" class = "fa fa-external-link"></i></a></label>
                    <input type="text" id="affiche-detenus" name="affiche-detenus" value="0">
                </div>

                <div>
                    <label>Affiches didacticiels
                        <a title="Aperçu" href="http://localhost/Annuaire_DSI/wp-content/uploads/2021/08/affiche_didactique.png" target="_blank" rel="noopener"><span>(60x80)</span>
                            <i style="font-size:12px; color:grey" class = "fa fa-external-link"></i></a></label>
                            <input type="text" id="affiche-didactique" name="affiche-didactique" value="0">
                    </div>

                    <div class="align">
                        <label>Affiches pour les familles
                        <a title="Aperçu" href="http://localhost/Annuaire_DSI/wp-content/uploads/2021/08/affiche_famille.png" target="_blank" rel="noopener"><span>(40x60)</span>
                            <i style="font-size:12px; color:grey" class = "fa fa-external-link"></i></a></label>
                            <input type="text" id="affiche-famille" name="affiche-famille" value="0">
                        </div>

                        <div>
                            <label>Flyers
                                <a title="Aperçu" href="http://localhost/Annuaire_DSI/wp-content/uploads/2021/08/flyer.png" target="_blank" rel="noopener"><span>(A4 plié en 3)</span>
                                    <i style="font-size:12px; color:grey" class = "fa fa-external-link"></i></a></label>
                                    <input type="text" id="flyers" name="flyers" value="0">
                            </div>

                            <div class="row">
                                <input type="submit" value="Envoyer">
                            </div>
                        </form>
                    </div>
                </body>
            </html>
```

```
html
{
    font-family: sans-serif;
    width: text-size-adjust: 100%;
    -ms-text-size-adjust: 100%;
    background-color: #f0f0f0;
    height: 100%;
}

/*body styles*/
body
{
    margin-left: auto;
    margin-right: auto;
    margin-top: 50px 40px;
    width: 470px;
    border: 1px solid black;
    background-color: white;
}

/*image style and position*/
img
{
    display: block;
    margin-left: auto;
    margin-right: auto;
    margin-bottom: 10px;
    width: 100px;
}

/*header style*/
h1
{
    font-size: 2em;
    margin: 0.6em 0;
}

/*change specific text colour*/
span
{
    color: #0000ff;
}

/*change colour on hover*/
span:hover
{
    color: #009900;
}

/*link style*/
a
{
    text-decoration: none;
}

/*rank (old display type) */
input, label
{
    display: block;
}

/*align items vertically*/
.align
{
    flex-direction: column;
    align-items: center;
    justify-content: space-between;
}

/*position text in row*/
.row
{
    font-size: 1px;
    margin-right: 20px;
}

/*input field style*/
input[type="text"], input[type="text"], input[type="text"]
{
    width: 240px;
}

/*input field style*/
input[type="button"]
{
    border: none;
    border-radius: 1px;
    color: #0000ff;
    font-family: sans-serif;
    font-size: 1px;
    font-weight: bold;
    line-height: 1px;
    padding: 10px;
    text-transform: uppercase;
    font-family: 'Ubuntu', sans-serif;
}

/*submit button style*/
input[type="submit"]
{
    border: none;
    border-radius: 1px;
    color: #fff;
    background-color: #0000ff;
    font-size: 1px;
    font-weight: bold;
    line-height: 1px;
    padding: 10px;
    text-transform: uppercase;
    font-family: 'Ubuntu', sans-serif;
}

/*submit button color on hover*/
input[type="submit"]:hover
{
    background: #222222;
}

/*submit button color on alive and focus*/
input[type="submit":focus], input[type="submit":active]
{
    background: #777777;
}

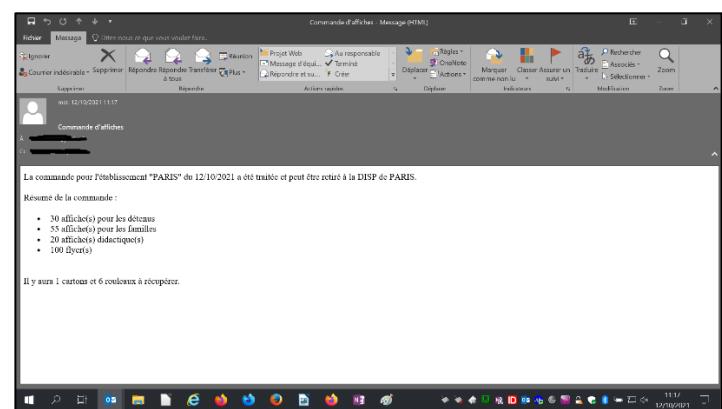
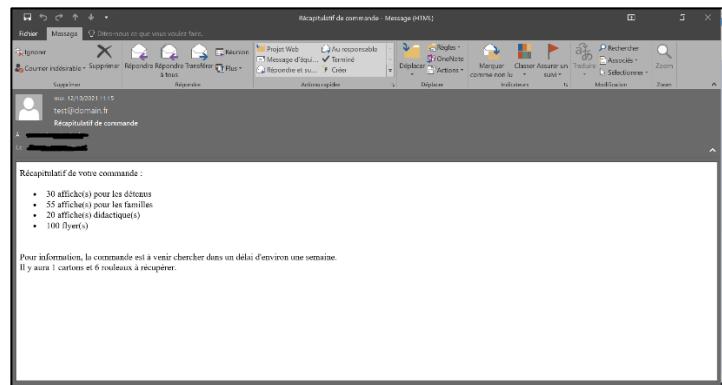
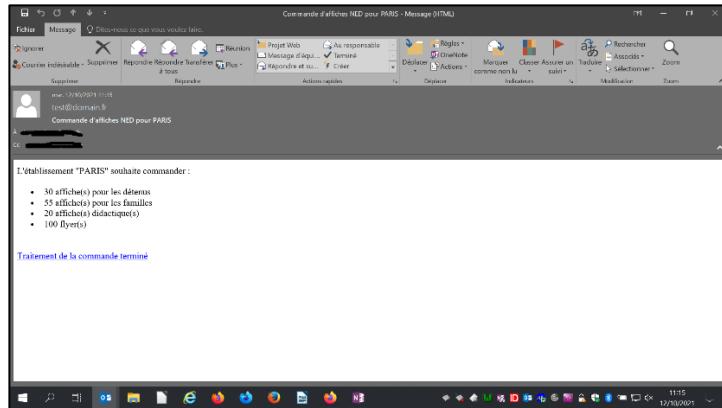
/*input text style on focus*/
input[type="text"]:-webkit-focus,
input[type="text"]:-moz-focus
{
    color: #000;
}

/*input text style on focus*/
input[type="text"]:-webkit-focus,
input[type="text"]:-moz-focus
{
    color: #000;
}
```

Le code du formulaire NED

J'ai ensuite implémenté l'envoi des mails. Après envoi du formulaire, un mail avec la commande est envoyé à la reprographie et un mail de récapitulatif est envoyé à l'utilisateur. Lorsque l'agent clique sur le lien placé dans le mail, l'utilisateur est informé que sa commande est prête à être retirée ainsi que du nombre de cartons et de rouleaux qui sont à récupérer. Les commandes sont également enregistrées dans une base de données.

Portefeuille d'activités professionnelles



L'échange des mails

+ Options	id	établissement	date_commande	nb_affiches_detenus	nb_affiches_famille	nb_affiches_didactiques	nb_flyers	statut
<input type="checkbox"/> Copier Supprimer	1	PARIS	12/10/2021	30	55	20	100	En attente
<input type="checkbox"/> Copier Supprimer	2	LYON	12/10/2021	50	50	0	0	Traité le 12/10/2021

La base de données associé aux commandes

```

<?php
//Connect to database
try
{
    $bdd = new PDO('mysql:host=localhost;dbname=php;charset=utf8', 'root', '');
    array(PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION);
}
catch (Exception $e)
{
    die('Erreur : ' . $e->getMessage());
}

//Prepare query
$req = $bdd->prepare("INSERT INTO commandes_affiches(établissement, date_commande,
nb_affiches_detenus, nb_affiches_famille, nb_affiches_didactiques, nb_flyers, statut) VALUES
(:établissement, :date_commande, :nb_affiches_detenus, :nb_affiches_famille, :nb_affiches_didactiques, :nb_flyers, :statut)");

//Bind variables
$établissement = strtroupper($_POST['établissement']);
$date_commande = date('d/m/Y');

$envoi = 1; POST($envoi);

$nb_affiches_detenus = $_POST['affiche-detenus'];
$nb_affiches_famille = $_POST['affiche-famille'];
$nb_affiches_didactiques = $_POST['affiche-didactique'];
$nb_flyers = $_POST['flyers'];

//Convert string to int
$nb_affiches_total = (int)$nb_affiches_detenus + (int)$nb_affiches_famille + (int)
$nb_affiches_didactiques;
$nb_flyers_total = (int)$nb_flyers;

//Calculate number of boxes and rolls
$nb_rouleaux = 1;
while($nb_affiches_total > 20)
{
    $nb_affiches_total -= 20;
    $nb_rouleaux += 1;
}

$nb_cartons = 1;
while($nb_flyers_total > 500)
{
    $nb_flyers_total -= 500;
    $nb_cartons += 1;
}

$statut = "En attente";

//Execute query, add data in database
$req->execute(array(
    ':établissement' => $établissement,
    ':date_commande' => $date_commande,
    ':nb_affiches_detenus' => $nb_affiches_detenus,
    ':nb_affiches_famille' => $nb_affiches_famille,
    ':nb_affiches_didactiques' => $nb_affiches_didactiques,
    ':nb_flyers' => $nb_flyers,
));
$statut => $statut;

//Get last ID
$response = $bdd->query("SELECT id FROM commandes_affiches ORDER BY id DESC LIMIT 1");
$array = $response->fetch(PDO::FETCH_ASSOC);
$last_id = implode($array);

//Convert array to string
$last_id = implode($array);

/////////////////////////////////////////////////////////////////Main mail
/////////////////////////////////////////////////////////////////

//Mail recipients
$to = "laura.huber@justice.fr"; //Primary recipient (repro disp paris@justice.fr)
$cce = "laura.huber@justice.fr";
//Additional recipients (unité fonctionnelle disp-paris@justice.fr)

//Completed link
$établissement = "localhost/Aenuairc_DSI/php/affiches/terminer.php?mail=$last_id";

//Mail message body
$message = "
<html>
<head>
    <title>$établissement souhaite commander :</title>
</head>
<body>
    <div>
        <li>$nb_affiches_detenus affiche(s) pour les détenus</li>
        <li>$nb_affiches_famille affiche(s) pour les familles</li>
        <li>$nb_affiches_didactiques affiche(s) didactique(s)</li>
        <li>$nb_flyers flyer(s)</li>
    </ul>
    <p><a href=\"$établissement\">Traitement de la commande terminé</a></p>
</body>
</html>
";
//Mail header content
$headers[] = "MIME-Version: 1.0";
$headers[] = "Content-type: text/html; charset=utf-8";
$headers[] = "From: $last_id";
$headers[] = "Cc: $cce";
//Site message on submit
echo "Votre message a été envoyé avec succès.";

//Send mail to $mail
mail($to, "Commande d'affiches MDP pour $établissement",
$message, implode("\r\n", $headers));

/////////////////////////////////////////////////////////////////Récap mail
/////////////////////////////////////////////////////////////////

//Recap mail recipient
$to_rec = $envoi; //User
//Recap mail message body
$message = "
<html>
<head>
    <title>Récapitulatif de votre commande :</title>
</head>
<body>
    <ul>
        <li>$nb_affiches_detenus affiche(s) pour les détenus</li>
        <li>$nb_affiches_famille affiche(s) pour les familles</li>
        <li>$nb_affiches_didactiques affiche(s) didactique(s)</li>
        <li>$nb_flyers flyer(s)</li>
    </ul>
    <p>Pour information, la commande est à venir chercher dans un délai d'environ une semaine
    Il y aura $nb_cartons cartons et $nb_rouleaux rouleaux à récupérer.</p>
</body>
</html>
";
//Mail header content
$headers[] = "MIME-Version: 1.0";
$headers[] = "Content-type: text/html; charset=utf-8";
$headers[] = "From: $to";
//Send mail to $to rec (user)
mail($to, "Récapitulatif de commande",
$message, implode("\r\n", $headers));

>
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="utf-8" />
</head>
</html>
";

```

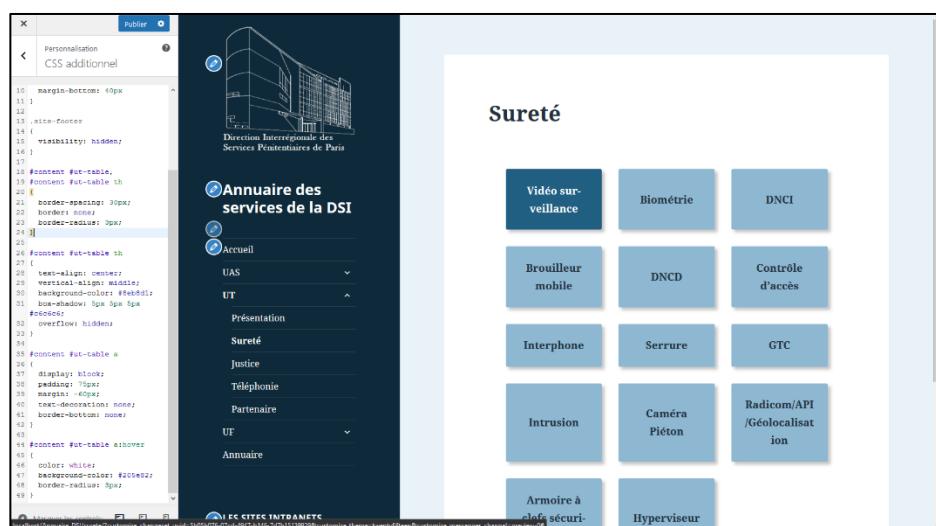
Le code PHP du formulaire qui gère l'envoi des mails et la connexion avec la base de données

Portefeuille d'activités professionnelles

Au cours de cette mission, j'ai contacté les différentes unités à plusieurs reprises afin de remplir l'annuaire avec les informations désirées et d'adapter le design et la structure selon leurs commentaires. En outre, je fais continuellement des petites modifications et corrections de la page.



La page d'accueil après avoir été retravaillé



Mise en œuvre du menu de l'onglet « unité technique »

Récemment, nous avons migré le site local vers un serveur en ligne à l'aide d'un plugin de migration. A l'heure actuelle, je travaille donc sur la version de production et sur une version test locale.

Actuellement, je m'occupe de la réalisation d'une fiche navette/fiche arrivant numérique. Pour cela, j'utilise TCPDF, une classe PHP pour créer des documents PDF ainsi que PHPMailer, une bibliothèque d'envoi de mails plus complète que la fonction mail() intégrée que j'utilisais jusqu'à présent et qui permet notamment l'envoi de pièces jointes. Après envoi de la fiche, les informations sont ajoutées dans une base de données et un PDF est créé qui sera envoyé par mail. Le but est que le PDF passe de service à service et se remplit au fur et à mesure.

Portefeuille d'activités professionnelles

DISP PARIS Département des Systèmes d'information	FORMULAIRE ARRIVANT D' MINISTÈRE DE LA JUSTICE
--	---

à délivrer au rapport du DI : parcours de l'enquête

Document destiné à l'information et à la consultation des personnes publiques (téléphonistes, fonctionnaires, agents administratifs, etc.) et à leur transmission à l'autorité compétente.

Attention : l'envoi régulier et régulièrement de l'enquête au rapport, qui n'a pas encore clôture dans le délai de 15 jours, entraîne une amende de 100 €.

NOM :

Prenom :

Secrétaire ou chargé de mission _____

Prénom :

Prénom :

Unité ou Service :

Situation :

Le DI est tenu de prendre en compte les demandes de création de compte pour APPF, liste de faire et d'écrire

Veuillez à l'heure de l'envoi du rapport du DI : ce document sera validé

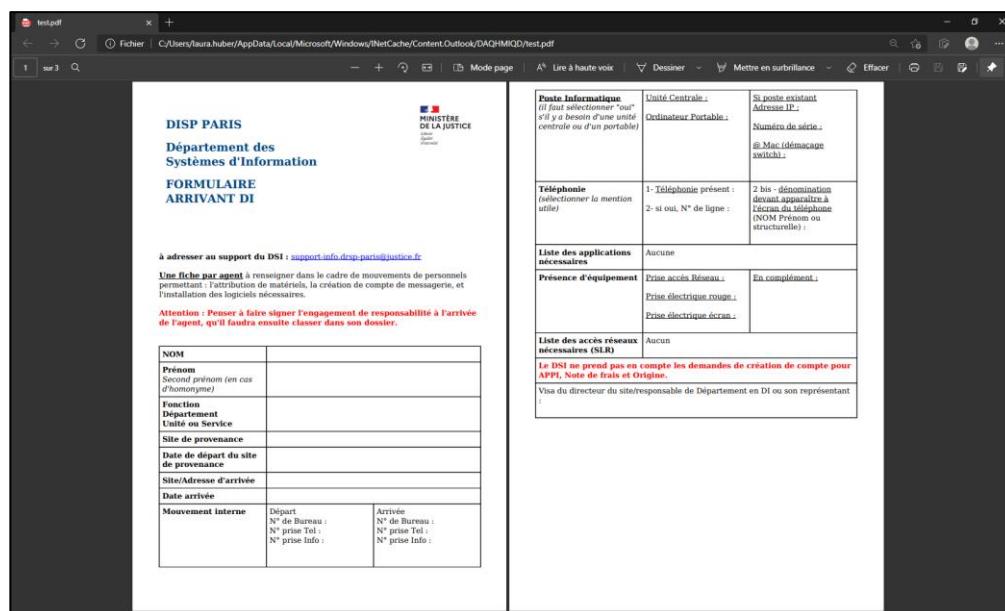
Objet de l'enquête :

Cadeau, don, aide

Cadeau, aide

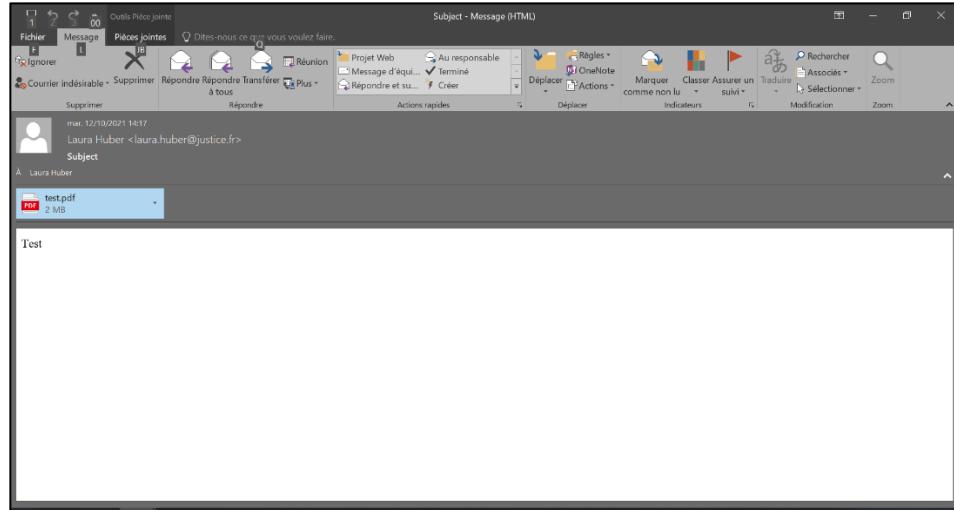
<div data-bbox="220 2786 270 27

Version préliminaire de la fiche navette



Aperçu du document PDF vierge

Portefeuille d'activités professionnelles



Mail avec pièce jointe

Le code PHP de la fiche navette

Compte rendu d'Activité		Fiche n° 2
Libellé court	Compte rendu mission entreprise	
Description	Compte rendu d'un projet de développement d'application.	
Contexte	Mission accomplie au sein de l'unité fonctionnelle de la DISP de Paris.	

Localisation	<input checked="" type="checkbox"/> Entreprise <input type="checkbox"/> CFA
Source	<input type="checkbox"/> 1ère année <input checked="" type="checkbox"/> 2ème année <input type="checkbox"/> TP <input type="checkbox"/> PPE
Cadre	<input checked="" type="checkbox"/> Seule <input type="checkbox"/> En équipe
Type	<input checked="" type="checkbox"/> Vécu <input type="checkbox"/> Observé <input type="checkbox"/> Simulé

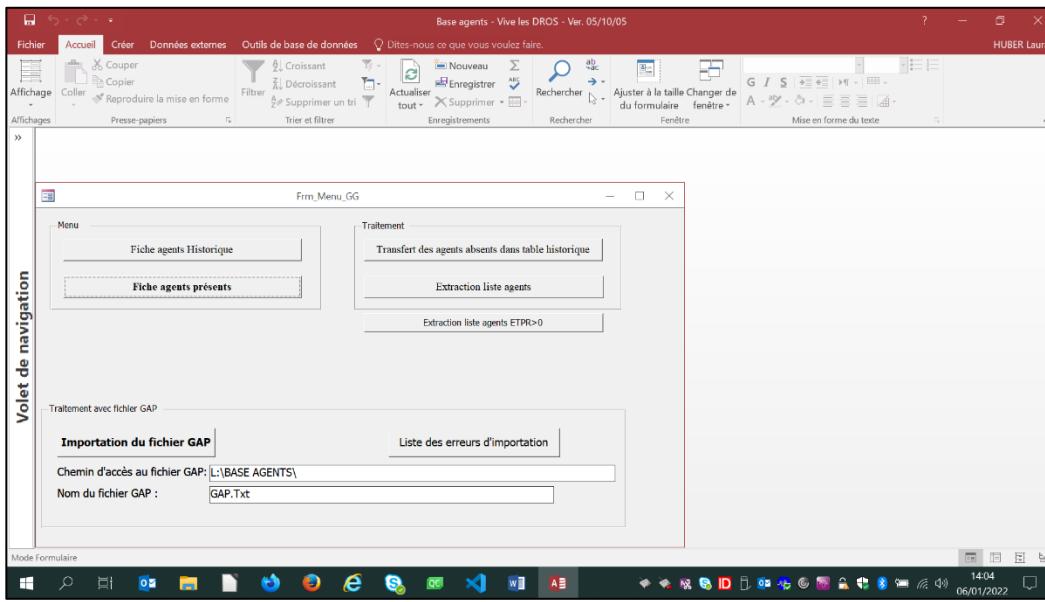
Environnement Technologique	(Qt), Microsoft Access, WAMP, phpMyAdmin
Moyens	(C++), HTML5, CSS, Bootstrap, PHP7, AJAX, MySQL
Avis personnel	Projet de développement qui m'a permis de me familiariser avec Qt et Access. Cependant, comme pour le premier projet, je n'ai pas eu d'aide sur le plan technique et j'ai donc mis plus longtemps que nécessaire, ce qui était démotivant.

Activités mises en œuvre pour la réalisation de cette situation	
	Organisation de son développement professionnel
	Conception et développement d'une solution applicative
	Maintenance corrective et évolutive d'une solution applicative
	Gestion des données

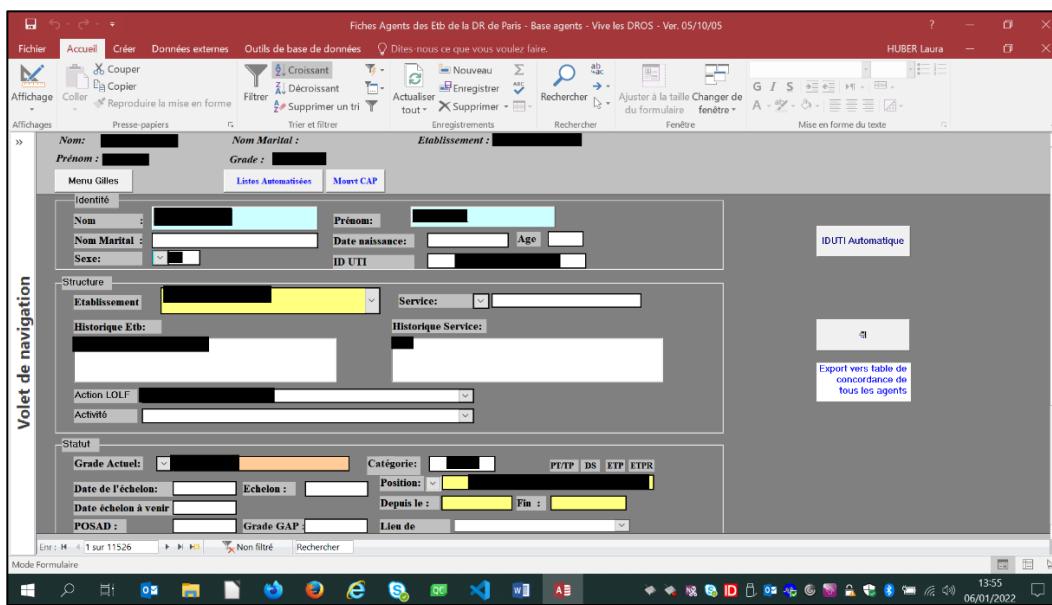
Déroulement de l'activité :

Ma mission consiste en la transformation d'une base de données Access en application de bureau.

La base de données en question contient les fiches des agents des établissements de Paris et doit être modernisée.



Le menu de la base des agents

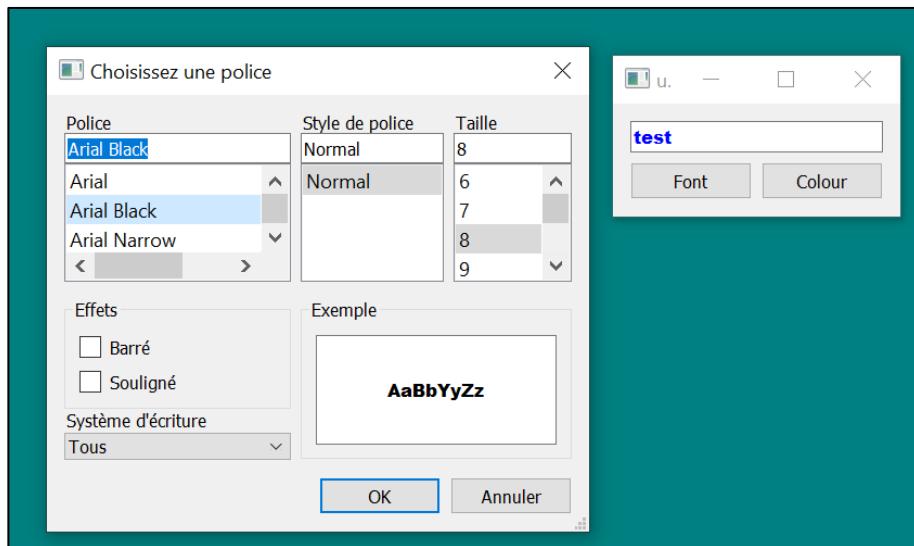


Une fiche agent dans la base de données Access

Pour commencer, il fallait choisir un langage et un framework adaptés à mes besoins. Après quelques heures de recherches, j'ai décidé d'utiliser C++ et Qt car cette dernière est multiplateforme et dispose d'une

documentation très complète et bien détaillée. En outre, grâce à la licence LGPL (GNU Lesser General Public License), elle peut être utilisée gratuitement. Un autre avantage est que Qt est, au départ, conçue pour être utilisée en C++, le langage avec lequel je suis le plus à l'aise.

Par la suite, j'ai donc procédé à l'installation et à la prise en main de l'IDE Qt Creator et j'ai créé une première fenêtre afin de me familiariser avec Qt.



Des premiers essais avec Qt (N. B. : les boutons « Font » et « Colour » envoient vers des widgets)

```
#ifndef DEF_MAFENetre
#define DEF_MAFENetre

#include <QApplication>
#include <QTranslator>
#include <QLocale>
#include <QLibraryInfo>
#include "MaFenetre.h"

class MaFenetre : public QWidget
{
    Q_OBJECT

public:
    void ouvrirFont();
    void ouvrirColour();

protected:
    QGridLayout *m_layout;
    QLineEdit *m_lineEdit;
    QPushButton *m_boutonFont;
    QPushButton *m_boutonColour;
};

#endif

#include "MaFenetre.h"

MaFenetre::MaFenetre() : QWidget()
{
    m_layout = new QGridLayout;
    m_lineEdit = new QLineEdit(this);
    m_lineEdit->move(100, 100);

    m_boutonFont = new QPushButton("Font", this);
    m_boutonFont->move(100, 150);

    m_boutonColour = new QPushButton("Colour", this);
    m_boutonColour->move(100, 200);

    m_layout->addWidget(m_lineEdit, 0, 0, 1, 2);
    m_layout->addWidget(m_boutonFont, 1, 0, 1, 1);
    m_layout->addWidget(m_boutonColour, 1, 1, 1, 1);

    this->setLayout(m_layout);

    QObject::connect(m_boutonFont, SIGNAL(clicked()), this, SLOT(ouvrirFont()));
    QObject::connect(m_boutonColour, SIGNAL(clicked()), this, SLOT(ouvrirColour()));
}

void MaFenetre::ouvrirFont()
{
    bool ok = false;

    QFont police = QFontDialog::getFont(ok, m_lineEdit->font(), this, "Choisissez une police");

    if (ok)
    {
        m_lineEdit->setFont(police);
    }
}

void MaFenetre::ouvrirColour()
{
    QColor couleur = QColorDialog::getColor(Qt::white, this);

    QPalette palette;
    palette.setColor(QPalette::Text, couleur);
    m_lineEdit->setPalette(palette);
}

int main(int argc, char *argv[])
{
    QApplication app(argc, argv);

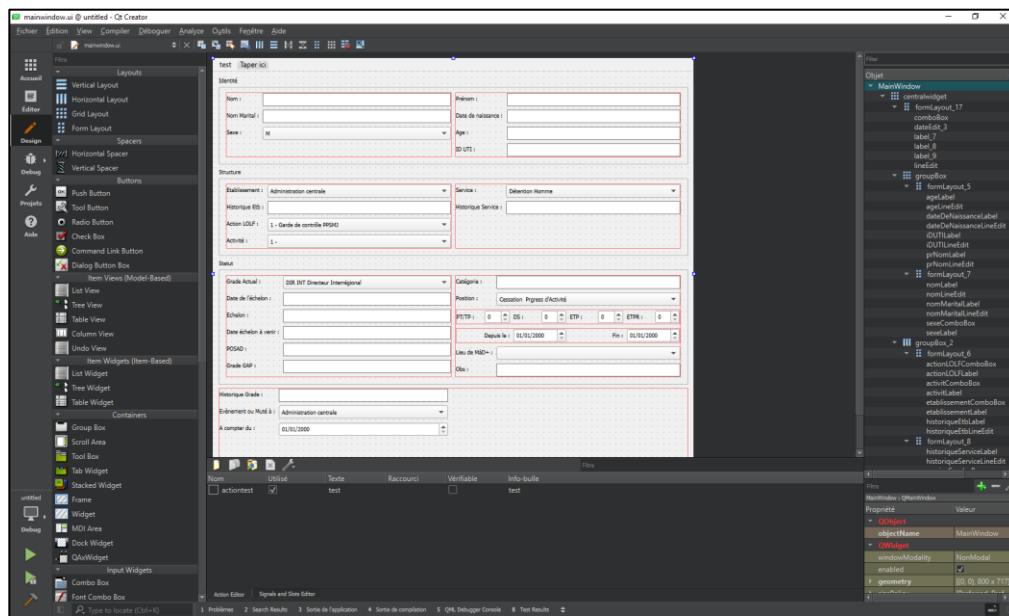
    QString locale = QLocale::system().name().section('_', 0, 0);
    QTranslator translator;
    translator.load(QString("qt_") + locale, QLibraryInfo::location(QLibraryInfo::TranslationsPath));
    app.installTranslator(&translator);

    MaFenetre fenetre;
    fenetre.show();

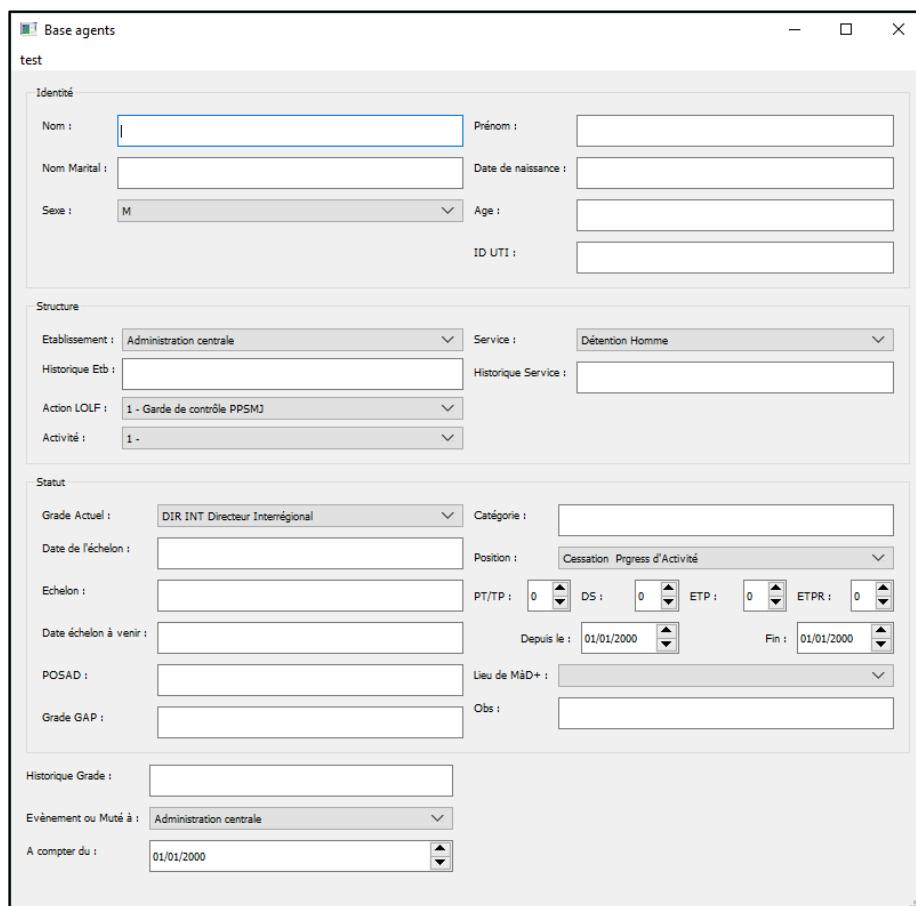
    return app.exec();
}
```

Les premiers bouts de code

Après, j'ai utilisé Qt Designer, une interface graphique qui permet de concevoir visuellement des interfaces, pour reproduire une fiche agent.



Une fiche agent dans Qt Designer



Une fiche agent une fois compilée

J'ai également exporté la base de données Access vers MySQL en la transformant en fichier texte et en l'important via la fonction « Import CSV using LOAD DATA » de phpMyAdmin.

La base de données dans phpMyAdmin

Finalement, après avoir pesé le pour et le contre, nous avons opté pour une application web plutôt qu'une application de bureau car elle ne nécessite pas d'installation sur le poste des agents et peut plus facilement être mise à jour. Par conséquent, j'ai abandonné le projet Qt en faveur d'une solution utilisant HTML, CSS et PHP ainsi qu'AJAX.

Pour commencer, j'ai transformé la fiche agent que j'avais préalablement créé avec Qt en formulaire HTML.

Une fiche agent en version web, les listes déroulantes sont alimentées par la base de données via un boucle PHP

Portefeuille d'activités professionnelles

Ensuite, j'ai repris le menu de la base Access en tant que barre de navigation visible sur toutes les pages de l'application. En outre, j'ai fait des ajustements au niveau de la taille et de la position des champs afin qu'ils soient plus faciles à modifier et j'ai ajouté un bouton « auto » qui génère l'ID UTI à partir des valeurs nom et prénom qui se trouvent dans les champs correspondants, en les séparant par une virgule. Ce dernier utilise un script AJAX pour cela.

Suite à un rendez-vous avec le responsable de la base de données, j'ai effectué d'autres modifications concernant les tables de la base et les champs du formulaire ainsi que le menu de navigation.

The screenshot shows a detailed form for managing agent records. The top navigation bar includes 'Menu', 'Traitement', 'Liste automatiques' (which is highlighted in blue), and 'Mouvement CAP'. The main title of the form is 'Fiche agents présents'. The form itself is organized into several sections:

- Identité:** Contains fields for Name (Nom), First Name (Prénom), Marital Status (Nom marital), Birth Date (Date de naissance), Age (Age), Sex (Sexe with dropdown F/M), and ID UTI (with an 'auto' button).
- Structure:** Includes fields for Establishment (Etablissement: PROVENANCE hors DR Paris), Service (Service: Déten...), Historical Establishment (Historique Etb), and Historical Service (Historique Service).
- Statut:** Contains fields for Current Grade (Grade actuel: DIR INT Directeur Interrégional), Category (Catégorie: PS), Position (Position dropdown), POSAD, Location of Appointment (Lieu de M&D: PROVENANCE hors DR Paris), Observations (Obs), and a grid for PT/TP, DS, ETP, and ETPR.
- Historique Grade:** A section for historical grade information.
- Evenement ou Muté à:** A dropdown for events or moves to another location (PROVENANCE hors DR Paris).
- Date:** An input field for the date 'A compter du: dd/mm/yyyy'.
- Action:** A blue 'Ajouter' (Add) button at the bottom right.

La page « Fiches Synoptiques » de l'application web avec le menu de navigation

Portefeuille d'activités professionnelles

Le code de la page « Fiches Synoptiques »

```

<?php
// Ouvrir la session, permet d'accéder à la variable &_SESSION
session_start();

if(isset($_POST['nomVal']) && isset($_POST['prenomVal']))
{
    //Initialisation des variables avec les valeurs des variables $_POST envoyés par ajax
    $nom = $_POST['nomVal'];
    $prenom = $_POST['prenomVal'];

    if(empty($nom) && empty($prenom))
    {
        $iduti = "";
    }
    else
    {
        //Si les variables sont initialisées, initialiser iduti avec leurs valeurs concaténées
        $iduti = $nom . ' ' . $prenom;
    }

    //Stockez iduti dans la variable $_SESSION
    $_SESSION['iduti'] = $iduti;
}
?>

```

Le code du script AJAX

De plus, j'ai mis en place un bouton qui permet d'ajouter une fiche à la base des agents.
J'ai également créé le frontend des formulaires de création de listes automatisées et de résultats des mouvements.

Le frontend du formulaire des listes automatisées

Le frontend du formulaire de résultats des mouvements

En ce moment, j'essaie de créer une barre de recherche avec pagination des résultats.

La barre de recherche et la pagination

```

<script>
$(document).ready(function()
{
    $("#myInput").keypress(function(event)
    {
        if(event.which == 13)
        {
            var value = $(this).val().toLowerCase();
            $("#fiches_synoptiques, #fiches_synoptiques form").filter(function()
            {
                $(this).toggle($(this).text().toLowerCase().indexOf(value) > -1)
            });
        }
    });
</script>

```

Le script pour la recherche

```

<!--Pagination-->
<nav id="pagination">
<ul class="pagination justify-content-center">
<?php

if($page == 1)
{
    echo "<li class='page-item disabled'><a class='page-link' href='?page=#' tabindex='-1'>Précédent</a></li>";
}
else
{
    $precedent = $page-1;
    echo "<li class='page-item active'><a class='page-link' href='?page=$precedent' tabindex='-1'>Précédent</a></li>";
}

echo "<li class='page-item'><a class='page-link' href='?page=1'>1</a></li>";

if($page > 5)
{
    echo "<li class='page-item disabled'><a class='page-link' href='?'>...</a></li>";
}

for($i=$page-5; $i<=$page+5; $i++) {
    while($i<2)
    {
        $i++;
    }
    if($page!=$i)
    {
        echo "<li class='page-item'><a class='page-link' href='?page=$i'>$i</a></li>";
    }
    else
    {
        echo "<li class='page-item active'><a class='page-link' href='?page=$i'>$i</a></li>";
    }
    if($i == $nb_agents-1)
    {
        break;
    }
}

if($page < $nb_agents-4)
{
    echo "<li class='page-item disabled'><a class='page-link' href='?'>...</a></li>";
}

if($page == $nb_agents)
{
    echo "<li class='page-item active'><a class='page-link' href='?page=$nb_agents'>$nb_agents</a></li>";
    echo "<li class='page-item'><a class='page-link' href='?page=#'>Suivant</a></li>";
}
else
{
    echo "<li class='page-item'><a class='page-link' href='?page=$nb_agents'>$nb_agents</a></li>";
    $suivant = $page+1;
    echo "<li class='page-item'><a class='page-link' href='?page=$suivant'>Suivant</a></li>";
}

?>
</ul>
</nav>

```

```
$sql_count = "SELECT COUNT(CodeAgent) AS total FROM fiches_agents_des_etb_de_la_dr_de_paris"; // Compter le nombre d'agents dans la table
$stmt_count = $pdo->query($sql_count);
$nb_row = $stmt_count->fetch(PDO::FETCH_ASSOC);
$nb_agents = ceil($nb_row["total"]); // Stocker le nombre d'agents dans la table

@$page = $_GET["page"]; // Ignorer les messages d'erreur générés par cette expression
if(empty($page) || $page>$nb_agents) // Si la page n'est pas définie ou est supérieure au nombre total des pages, envoyer sur la page 1
{
    $page = 1;
}
$elements_par_page = 1;
$debut = ($page-1)*$elements_par_page;
```

Le code de la pagination

Compte rendu d'Activité		Fiche n° 3
Libellé court	Compte rendu Projet Personnel Encadré	
Description	Compte rendu de la mise en place d'un MédiaWiki.	
Contexte	Activité réalisée dans le cadre du PPE 3 à la Fdme de Massy.	

Localisation	<input type="checkbox"/> Entreprise <input checked="" type="checkbox"/> CFA
Source	<input type="checkbox"/> 1ère année <input type="checkbox"/> 2ème année <input type="checkbox"/> TP <input checked="" type="checkbox"/> PPE
Cadre	<input type="checkbox"/> Seule <input checked="" type="checkbox"/> En équipe
Type	<input checked="" type="checkbox"/> Vécu <input type="checkbox"/> Observé <input type="checkbox"/> Simulé

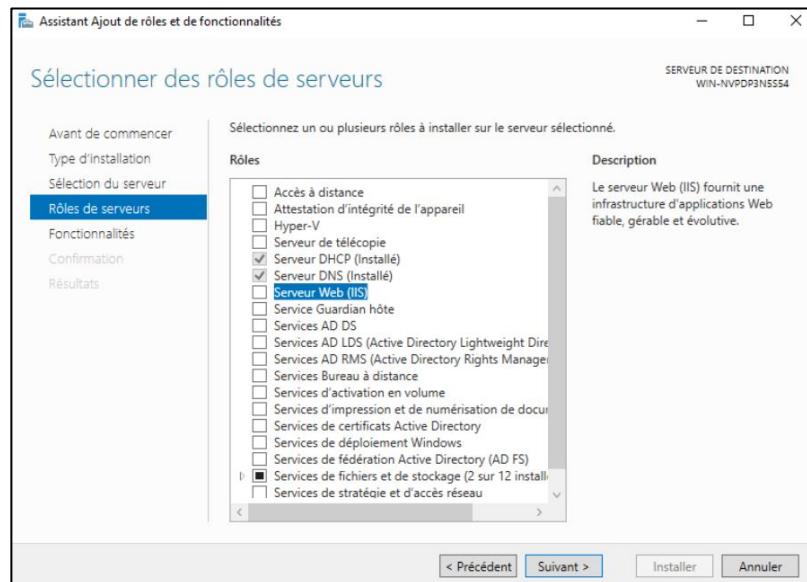
Environnement Technologique	Windows Server, Debian
Moyens	MySQL, MariaDB, Apache, PHP
Avis personnel	Comme ce projet était un travail de groupe, je le trouvais très difficile car cela demande une bonne communication.

Activités mises en œuvre pour la réalisation de cette situation	
	Mise à disposition des utilisateurs d'un service informatique
	Conception et développement d'une solution applicative
	Organisation de son développement professionnel

Déroulement de l'activité :

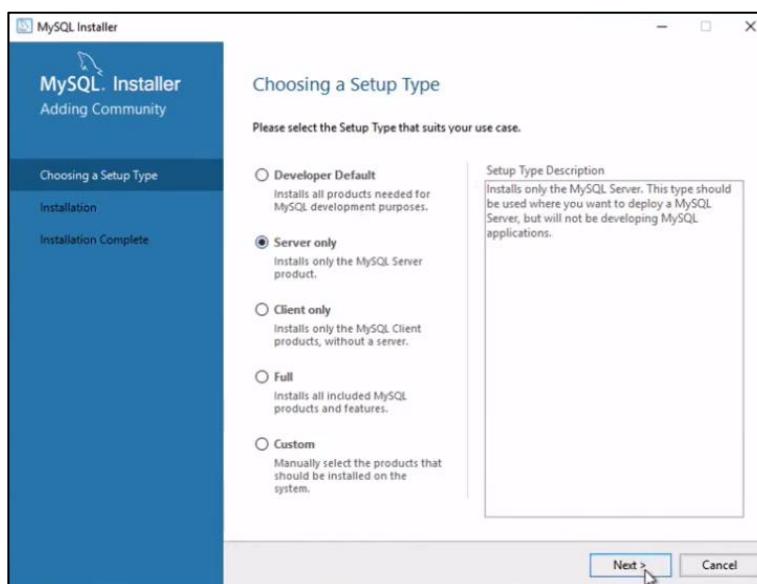
Dans le contexte du PPE 3, nous devions mettre en place un MédiaWiki sur « Les métiers de l'informatique », le sujet du PPE 1. Ce projet était à réaliser en groupe et les tâches à répartir de façon équilibrée. Afin que le MédiaWiki fonctionne sur Windows ainsi que sur Linux, il fallait qu'on l'installe et qu'on le configure dans les deux environnements. L'utilisation d'un intégrateur de solution n'était pas permise.

Nous avons débuté par l'installation de la version adaptée de MédiaWiki pour Windows. Pour cela, nous avons installé Windows Server et nous y avons mis en place un serveur web IIS.



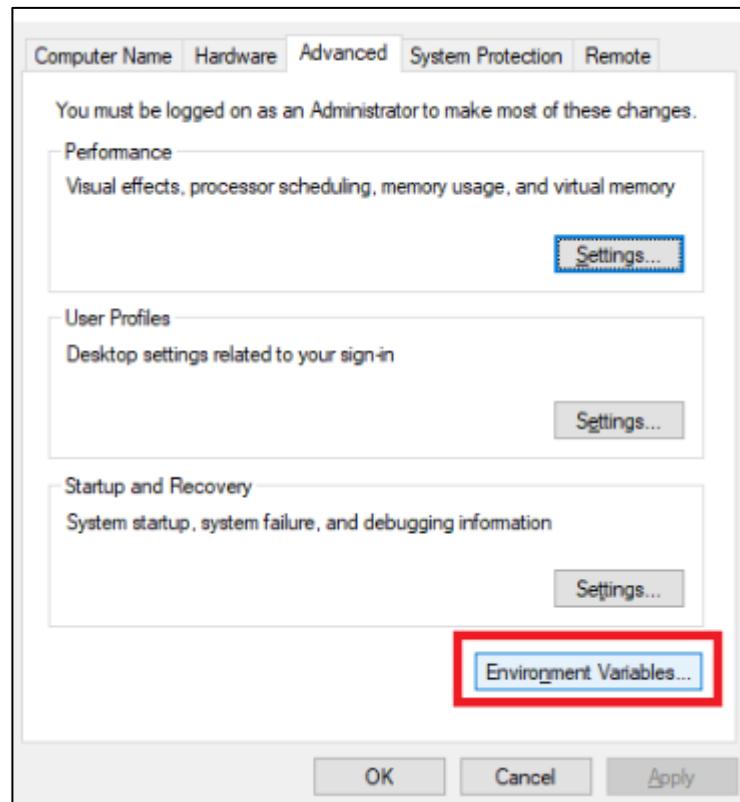
L'installation d'un serveur IIS

Ensuite, nous avons installé le MySQL Community Server pour la base de données qui sera associé au MédiaWiki.



L'installation de MySQL

Afin d'assurer le bon fonctionnement de notre MédiaWiki, nous avons également dû installer PHP ainsi que la librairie Visual C++ requise. L'étape suivante consistait à déplacer le contenu de l'archive contenant le fichier d'installation PHP vers le disque dur principal et à ajouter son chemin d'accès à la variable PATH.

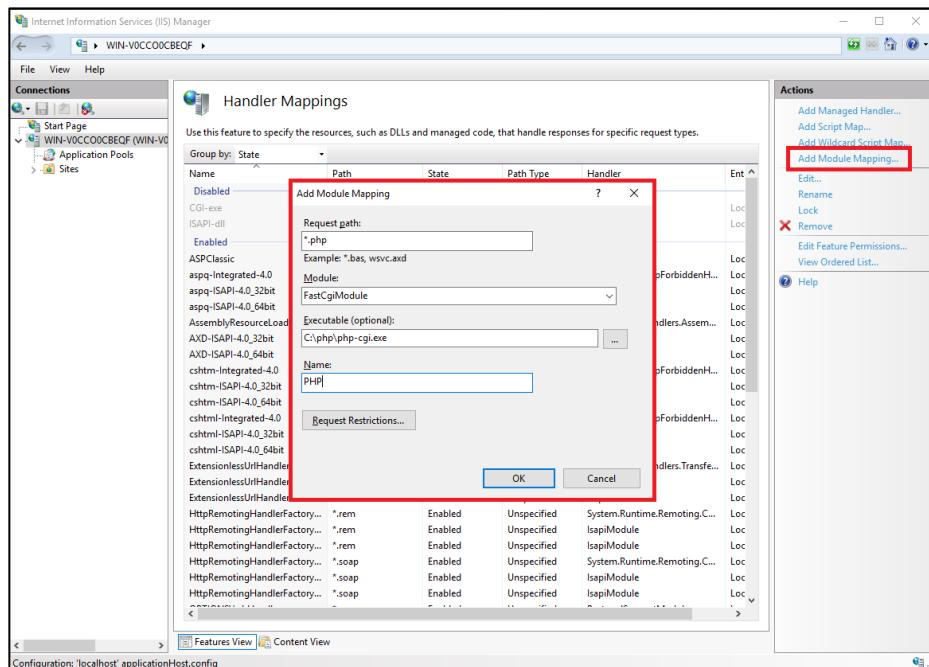


Modification de la variable PATH système

Après avoir modifié le fichier php.ini, nous avons ajouté le module PHP à notre serveur IIS.

```
extension=php_mysql.dll  
extension=php_mysqli.dll  
extension=php_gd2.dll  
extension=php_intl.dll  
extension=php_curl.dll  
extension=php_ldap.dll
```

Les modules à décommenter dans php.ini



Le module PHP

Une fois que nous avons préparé une base de données pour stocker les données des utilisateurs, nous avons procédé à l'installation et la configuration de cette dernière.



L'installation du MédiaWiki

Par la suite, nous avons mis en place un serveur Debian avec un serveur web Apache afin d'installer MédiaWiki dans un environnement Linux.

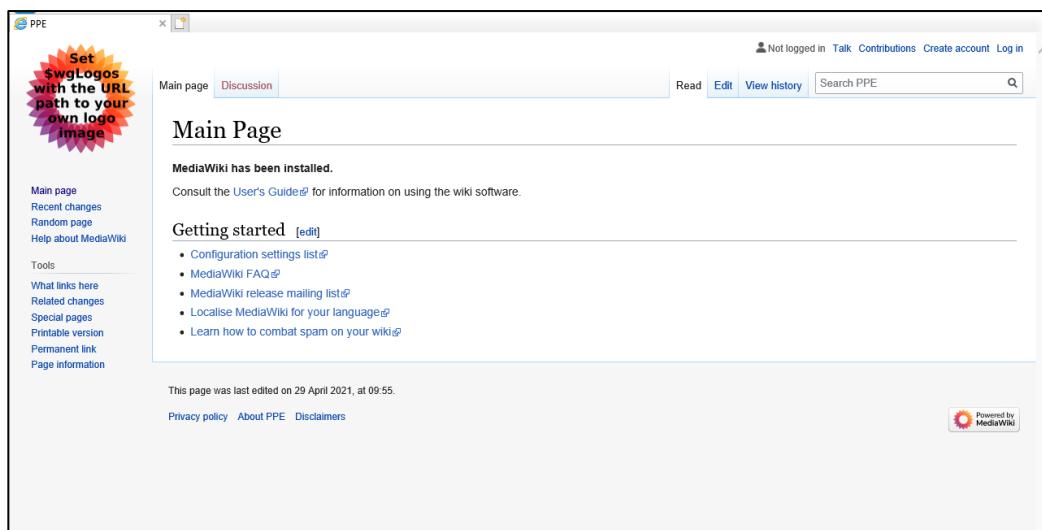
```
sudo apt-get install apache2 mariadb-server php php-mysql libapache2-mod-php
php-xml php-mbstring
```

La commande pour installer les paquets nécessaires pour un serveur Apache

Après l'installation des paquets, nous avons téléchargé MédiaWiki sur l'emplacement par défaut des sites web Apache et nous avons créé et configuré une base de données. La prochaine étape était la configuration de PHP, notamment l'augmentation de « upload_max_filesize » et de « memory_limit » dans le php.ini.

Pour terminer, nous avons installé et configuré MédiaWiki de la même façon que sur Windows.

Une fois l'installation terminée, il ne restait plus qu'à remplir le wiki avec le contenu du PPE 1 sur les métiers de l'informatique.



La page d'accueil du MédiaWiki

Afin de changer le logo par défaut, nous avons remplacé la variable \$wgLogos dans le fichier LocalSettings.php téléchargé après l'installation de MédiaWiki par une variable appelée \$wgLogo. Par la suite, nous avons téléchargé notre logo sur l'emplacement html d'Apache et modifié notre variable afin d'y mettre notre logo.

```
## The URL paths to the logo. Make sure you change this from the default,
## or else you'll overwrite your logo when you upgrade!
#$wgLogos = [ '1x' => "$wgResourceBasePath/resources/assets/wiki.png" ];
$wgLogo = "http://www.ppe.local/wiki.png";
```

Les modifications dans LocalSettings.php qui permettent d'afficher notre propre logo

The screenshot shows a MediaWiki site's main page. At the top, there is a navigation bar with links for 'Non connecté', 'Discussion', 'Contributions', 'Cr...', 'Accueil' (highlighted in blue), 'Discussion' (highlighted in red), 'Lire', 'Modifier', 'Voir l'historique', and a 'Re...' button. On the left side, there is a sidebar with links for 'Accueil', 'Modifications récentes', 'Page au hasard', 'Aide concernant MediaWiki', 'Outils', 'Pages liées', 'Suivi des pages liées', 'Pages spéciales', 'Version imprimable', 'Lien permanent', and 'Informations sur la page'. The main content area has a heading 'Accueil' and a section titled 'MediaWiki a été installé.' with a link to the 'Guide de l'utilisateur'. Below this, there is a section titled 'Pour démarrer' with a 'modifier' link and a bulleted list of links: 'Liste des paramètres de configuration', 'Questions courantes sur MediaWiki', 'Liste de discussion sur les distributions de MediaWiki', 'Adaptez MediaWiki dans votre langue', and 'Apprendre comment lutter contre le pourriel dans votre wiki'. At the bottom of the page, there is a note about the last modification and links to 'Politique de confidentialité', 'À propos de PPE FAJL', and 'Avertissements'.

La page d'accueil avec logo personnalisé

Pour la gestion des droits, nous avons modifié la variable \$wgGroupPermissions dans LocalSettings.php pour les utilisateurs non connectés ('*'), les utilisateurs connectés ('user') et les administrateurs ('sysop'). Ces derniers sont les seuls à avoir tous les droits tandis que les utilisateurs connectés et non connectés n'ont qu'un droit de lecture.

```
$wgGroupPermissions['*']['createaccount'] = true; #Les utilisateurs peuvent
créer un compte
$wgGroupPermissions['*']['read'] = true; #Ils peuvent lire
$wgGroupPermissions['*']['edit'] = false; #Ne peuvent pas modifier
$wgGroupPermissions['*']['createpage'] = false; #Ne peuvent pas créer de
page
$wgGroupPermissions['*']['move'] = false; #Ne peuvent pas déplacer
les pages
$wgGroupPermissions['*']['createtalk'] = false; #Ne peuvent pas créer de
pages de discussions
$wgGroupPermissions['*']['upload'] = false; #Ne peuvent pas poster des
images
$wgGroupPermissions['*']['reupload'] = false; #Ne peuvent pas réécrire
des images
$wgGroupPermissions['*']['minoredit'] = false; #Ne peuvent pas éditer les
pages et les marquer comme "éditions mineures"
```

Droits des utilisateurs non connectés

```
$wgGroupPermissions['user']['read'] = true; #Les utilisateurs connectés peuvent lire
$wgGroupPermissions['user']['edit'] = false; #Ne peuvent pas modifier
$wgGroupPermissions['user']['createpage'] = false; #Ne peuvent pas créer de page
$wgGroupPermissions['user']['move'] = false; #Ne peuvent pas déplacer les pages
$wgGroupPermissions['user']['createtalk'] = false; #Ne peuvent pas créer de pages de discussions
$wgGroupPermissions['user']['upload'] = false; #Ne peuvent pas poster des images
$wgGroupPermissions['user']['reupload'] = false; #Ne peuvent pas réécrire des images
$wgGroupPermissions['user']['minoredit'] = false; #Ne peuvent pas éditer les pages et les marquer comme "éditions mineures"
```

Droits des utilisateurs connectés

```
$wgGroupPermissions['sysop']['read'] = true;
$wgGroupPermissions['sysop']['edit'] = true;
$wgGroupPermissions['sysop']['move'] = true;
$wgGroupPermissions['sysop']['createpage'] = true;
$wgGroupPermissions['sysop']['createtalk'] = true;
$wgGroupPermissions['sysop']['upload'] = true;
$wgGroupPermissions['sysop']['reupload'] = true;
$wgGroupPermissions['sysop']['reupload-shared'] = true;
$wgGroupPermissions['sysop']['minoredit'] = true;
```

Droits des administrateurs

Enfin, nous avons également rédigé un manuel destiné à l'utilisateur ainsi que deux manuels techniques pour les différentes installations.

Procédure Utilisation du Mediawiki

Sommaire :

- I - Ordinateur sous Windows
- II - Ordinateur sous Linux

I - Ordinateur sous Windows

1) Prérequis :

- **Adresse IP et nom du Mediawiki** (fournie par le service informatique de votre entreprise)
- **Être sur le lieu de travail**

2) Configuration sur votre ordinateur

Le manuel d'utilisateur

Procédure Installation MediaWiki sur Serveur

Sommaire :

- I - Procédure sur Windows Server
- II - Procédure sur Linux Server

I - Procédure sur Windows Server

1) Prérequis :

- Connexion Internet (pour le téléchargement des outils nécessaires)
- Accès au serveur windows

2) Installation :

Le manuel technique

Compte rendu d'Activité		Fiche n° 4
Libellé court	Compte rendu Projet Personnel Encadré E5	
Description	Compte rendu d'un projet de développement d'un site de bibliothèque.	
Contexte	Mission accomplie dans le cadre de la formation	

Localisation	<input type="checkbox"/> Entreprise <input checked="" type="checkbox"/> CFA
Source	<input type="checkbox"/> 1ère année <input type="checkbox"/> 2ème année <input type="checkbox"/> TP <input checked="" type="checkbox"/> PPE
Cadre	<input checked="" type="checkbox"/> Seule <input type="checkbox"/> En équipe
Type	<input checked="" type="checkbox"/> Vécu <input type="checkbox"/> Observé <input type="checkbox"/> Simulé

Environnement Technologique	Visual Studio Code, WAMP, phpMyAdmin
Moyens	HTML5, CSS, Bootstrap, PHP, JavaScript, MySQL, GitHub
Avis personnel	Projet de développement web qui m'a permis d'approfondir mes connaissances et qui a un fort potentiel d'évolution.

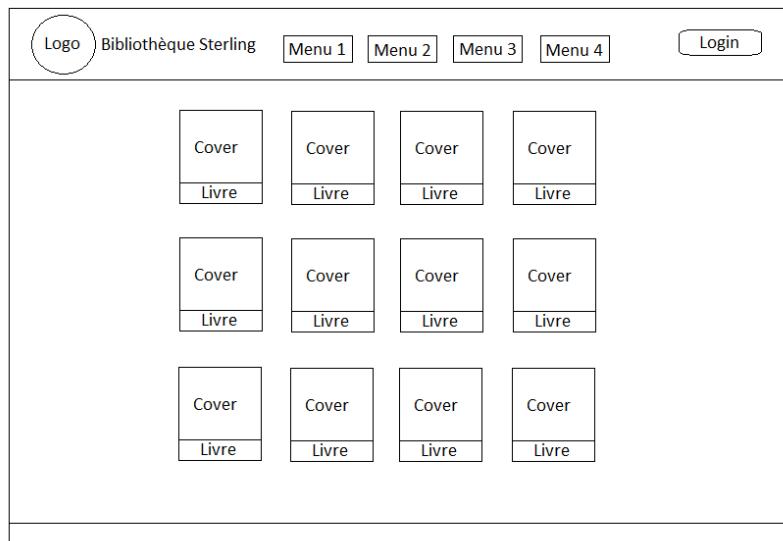
Activités mises en œuvre pour la réalisation de cette situation	
	Organisation de son développement professionnel
	Conception et développement d'une solution applicative
	Maintenance corrective et évolutive d'une solution applicative
	Gestion des données

Déroulement de l'activité :

Dans le contexte de la M2L (Maison des Ligues de Lorraine), j'ai mis en place une application en client léger, plus particulièrement un site web, la "bibliothèque Sterling", qui permet des télécharger des livres numériques en format PDF.

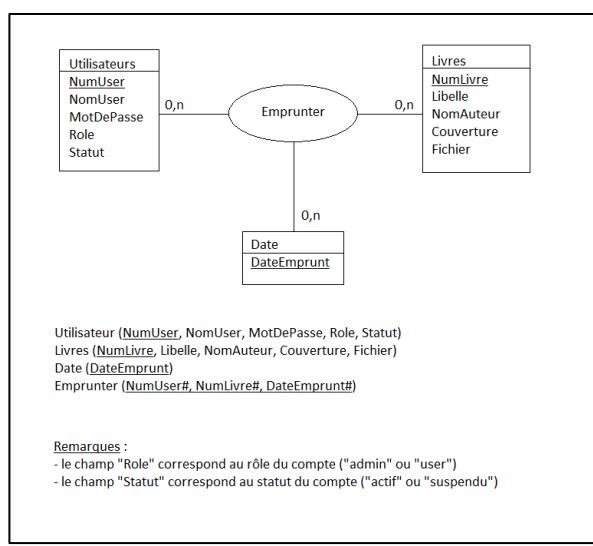
Tout d'abord, j'ai choisi les langages adaptés à mes besoins. J'ai décidé d'utiliser HTML et CSS pour le frontend, en utilisant le framework Bootstrap afin de gagner du temps et d'assurer une comptabilité avec la plupart des navigateurs, mais aussi parce que sa documentation est très complète. En outre, j'ai déjà utilisé Bootstrap au cours d'une autre mission et j'avais donc pu me familiariser avec celui-ci. Pour le backend, je me suis servie de PHP et de JavaScript et pour la base de données j'ai opté pour MySQL.

Pour commencer, j'ai créé une maquette simple afin d'avoir une idée de la position des différents éléments.



La maquette de la page d'accueil

J'ai également réalisé le MCD (Modèle Conceptuel des Données) et le MLD (Modèle Logique des Données) des tables dont j'aurai besoin dans la base de données.



Le MCD et le MLD de la base de données

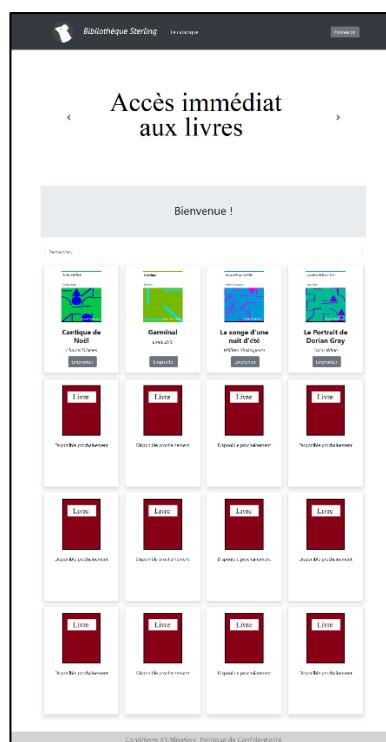
Ensuite, j'ai installé WampServer et j'ai mis en place la base de données MySQL dans phpMyAdmin.

The screenshot shows the phpMyAdmin interface for a MySQL database named 'bibliothèque'. The sidebar on the left lists various databases. The main panel shows a table of three rows: 'emprunter', 'livres', and 'utilisateurs'. Below the table, there are buttons for 'Print' and 'Data dictionary', and a 'Create table' form with fields for 'Name' and 'Number of columns' (set to 4). At the bottom, there's a 'Console' section.

La base de données dans phpMyAdmin

Avant de me lancer dans le projet, j'ai également installé l'application de bureau de GitHub pour la gestion de versions avec Git.

La première page que j'ai mise en place est la page d'accueil qui affiche le catalogue. Au début, j'avais inséré les livres en dur dans le code mais j'ai changé cela par la suite afin de stocker les livres dans la base de données. J'ai également créé la barre de menu et le bas de page avec les conditions d'utilisation et la politique de confidentialité ainsi qu'un carrousel qui affiche du texte.



Le catalogue

```
<!--ON PARCOUR LES LIGNES DE LA TABLE LIVRES DE LA BDD-->
<?php while($row = $stmt->fetch(PDO::FETCH_ASSOC)) : ?>
<div class="col-auto card shadow">
    <img src=<?php echo htmlspecialchars($row['Couverture']); ?>" class="card-img-top" alt="Cover" />
    <div class="card-body">
        <h1><?php echo htmlspecialchars($row['Libelle']); ?></h1>
        <p><?php echo htmlspecialchars($row['NomAuteur']); ?></p>
        <!--OUVRE LE FICHIER SI UTILISATEUR CONNECTE, RENVOIE SUR PAGE LOGIN LE CAS ECHEANT-->
        <?php
            if(!isset($_SESSION["loggedin"]) || $_SESSION["loggedin"] != true)
            {
                echo
                    '<a href="Pages/login.php">' .
                    '<button type="button" class="btn btn-secondary">Emprunter</button>' .
                    '</a>';
            }
            // Si l'utilisateur a le rôle admin, bouton emprunter et remplacé par bouton supprimer
            elseif(isset($_SESSION["loggedin"]) && $_SESSION["role"] == 'admin')
            { ?>
                <form method="post" action="Pages/SuppressionLivre.php">
                    <button type="submit" class="btn btn-danger" name="id" value=<?php echo htmlspecialchars($row['id']); ?>">
                        onclick="return confirm('Êtes-vous sûr de vouloir supprimer ce livre ?')">Supprimer</button>
                    </form>
            <?php }
            else
            { ?>
                <a href=<?php echo htmlspecialchars($row['Fichier']); ?>" download target="_blank">
                    <button type="button" class="btn btn-info">Emprunter</button>
                </a>
            <?php }
            ?>
        </div>
    </div>
</div>
<?php endwhile; ?>
```

Récupération des livres stockés dans la base de données

J'ai ensuite ajouté un formulaire de création de compte ainsi qu'un formulaire de connexion.

Nom d'utilisateur

Mot de passe

Confirmation du mot de passe

S'inscrire

Vous avez déjà un compte ? [Se connecter](#).

Le formulaire de création de compte

```

<!--Formulaire-->
<div class="container shadow" id="login">
    <form action=<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>" method="post">
        <!--Appeille le script (self) après submit-->
        <div class="form-group">
            <label for="username">Nom d'utilisateur</label>
            <input type="username" class="form-control <?php echo (!empty($username_err)) ? 'is-invalid' : '' ; ?>" value=<?php echo $username; ?>" id="username" name="username">
            <span class="invalid-feedback"><?php echo $username_err; ?></span>
        </div>
        <div class="form-group">
            <label for="password">Mot de passe</label>
            <input type="password" class="form-control <?php echo (!empty($password_err)) ? 'is-invalid' : '' ; ?>" value=<?php echo $password; ?>" id="password" name="password">
            <span class="invalid-feedback"><?php echo $password_err; ?></span>
        </div>
        <div class="form-group">
            <label for="confirm_password">Confirmation du mot de passe</label>
            <input type="password" class="form-control <?php echo (!empty($confirm_password_err)) ? 'is-invalid' : '' ; ?>" value=<?php echo $confirm_password; ?>" id="confirm_password" name="confirm_password">
            <span class="invalid-feedback"><?php echo $confirm_password_err; ?></span>
        </div>
        <button type="submit" class="btn btn-primary">Créer</button>
    </form>
</div>

```

Le code du formulaire de création de compte

Nom d'utilisateur

Mot de passe

Connexion

Vous n'avez pas encore de compte ? [Créez un compte.](#)

Le formulaire de connexion

```
<form action=<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>" method="post">
<!--Appelle le script (self) après submit-->
    <div class="form-group">
        <label for="username">Nom d'utilisateur</label>
        <input type="username" id="username" name="username" class="form-control"
            <?php echo (!empty($username_err)) ? 'is-invalid' : ''; ?>" value="<?php echo $username; ?>">
        <span class="invalid-feedback"><?php echo $username_err; ?></span>
    </div>
    <div class="form-group">
        <label for="password">Mot de passe</label>
        <input type="password" id="password" name="password" class="form-control"
            <?php echo (!empty($password_err)) ? 'is-invalid' : ''; ?>" value="<?php echo $password; ?>">
        <span class="invalid-feedback"><?php echo $password_err; ?></span>
    </div>
    <button type="submit" class="btn btn-primary">Connexion</button>
    <p><br>Vous n'avez pas encore de compte ? <a href="register.php">Créer un compte</a>.</p>
</form>
```

Le code du formulaire de connexion

Enfin, j'ai ajouté des rôles. Selon le rôle associé au compte, les fonctionnalités du site changent. Le rôle correspondant au compte connecté est stocké dans une variable globale afin de vérifier à quelles pages et fonctionnalités l'utilisateur peut accéder. Cela empêche également la manipulation d'URL.

```
// Ouvrir la session, permet d'accéder à la variable &_SESSION
session_start();

// Vérifier si l'utilisateur est connecté et a le rôle admin, redirection vers page login le cas échéant
if(!isset($_SESSION["loggedin"]) && $_SESSION["role"] !== 'admin')
{
    header("location: login.php");
    exit;
}
```

Vérification du rôle du compte

Pour les comptes "admin", j'ai mis en place un formulaire qui permet d'ajouter des livres au catalogue. La base de données stocke ensuite le chemin vers l'image de la couverture et vers le fichier pdf du livre et une copie des deux est ajoutée à un répertoire destiné à cet effet créé après envoi du formulaire.

Titre

Auteur

Image de couverture
 No file selected.

Fichier
 No file selected.

[Ajouter](#) [Annuler](#)

Le formulaire pour ajouter un livre

En outre, j'ai mis en œuvre une page de gestion des utilisateurs pour les comptes "admin" qui leur permet de voir les utilisateurs existants, de créer et de supprimer des comptes et de modifier le nom d'utilisateur, le mot de passe et le rôle associé à un compte. Ils peuvent également suspendre des comptes. Les comptes suspendus ne se peuvent plus connecter jusqu'au moment où ils sont réactivés.

ID	Nom d'utilisateur	Mot de passe	Rôle	Statut	Action
1	admin	\$2y\$10\$/ZbcdcdQ3RJyXJCgqJ1KGhO2dowlnb9cODjaMiF7GSDxgrRxJc.9ay	admin	actif	
15	test	\$2y\$10\$2uoO9WEHx8s235xAcqB.ujqAd76minHGRS.yXdRnHEWjZlpCEQKq	user	actif	
19	User	\$2y\$10\$718r5d.k0lwZZYfcVDDVeXStZgOzL3p1V4GT5CiKs23tsfA8akHe	user	actif	
17	test2	\$2y\$10\$KDHB.QrlThK1G7XHduAbW.tFouRZzdFSlessP3Z2yWCiiKEEnKd.5W	user	actif	
20	VotreNom	\$2y\$10\$sdC2kVPKsdW20d.TQUvVe.1e1mu6KAw.V1P70KBEF.0DgdkWa3A2e	user	actif	

Rechercher... [Créer un compte](#)

Conditions d'Utilisation Politique de Confidentialité

La page de gestion des utilisateurs

```
<?php
// Ouvrir la session, permet d'accéder à la variable &_SESSION
session_start();

// Inclure le fichier config
require_once "../config.php";

// Définition et initialisation des variables
$id = $_POST['id'];

// Préparation d'une requête delete
$sql = 'DELETE FROM utilisateurs WHERE id = :id';
$stmt = $pdo->prepare($sql);

// Liaison des variables à la requête en tant que paramètres
$stmt->bindParam(':id', $id, PDO::PARAM_INT);

// Exécution de la requête préparée
if ($stmt->execute())
{
    // Redirection vers page gestion des utilisateurs
    header('Location: GestionUser.php');
    exit();
}

// Fermer la requête
unset($stmt);
}

// Fermer la connexion
unset($pdo);
?>
```

Code pour la suppression d'un compte

```

<?php
// Ouvrir la session, permet d'accéder à la variable &_SESSION
session_start();

// Inclure le fichier config
require_once "../config.php";

// Définition et initialisation des variables
$id = $_POST['id'];

// Préparation d'une requête select
$sql = 'SELECT id FROM utilisateurs WHERE statut = "actif" AND id = :id';

if($stmt = $pdo->prepare($sql))
{
    // Liaison des variables à la requête en tant que paramètres
    $stmt->bindParam(':id', $id, PDO::PARAM_STR);

    // Exécution de la requête préparée
    if($stmt->execute())
    {
        // Mettre à jour le statut en fonction du statut actuel
        if($stmt->rowCount() == 1) // Retourne le nombre de ligne affecté par la dernière requête
        {
            // Préparation d'une requête update
            $sql = 'UPDATE utilisateurs SET statut = "suspendu" WHERE id = :id';
            $stmt = $pdo->prepare($sql);

            // Liaison des variables à la requête en tant que paramètres
            $stmt->bindParam(':id', $id, PDO::PARAM_INT);

            // Exécution de la requête préparée
            if ($stmt->execute())
            {
                // Redirection vers page gestion des utilisateurs
                header('Location: GestionUser.php');
                exit();
            }

            // Fermer la requête
            unset($stmt);
        }
        else
        {
            // Préparation d'une requête update
            $sql = 'UPDATE utilisateurs SET statut = "actif" WHERE id = :id';
            $stmt = $pdo->prepare($sql);

            // Liaison des variables à la requête en tant que paramètres
            $stmt->bindParam(':id', $id, PDO::PARAM_INT);

            // Exécution de la requête préparée
            if ($stmt->execute())
            {
                // Redirection vers page gestion des utilisateurs
                header('Location: GestionUser.php');
                exit();
            }

            // Fermer la requête
            unset($stmt);
        }
    }
    // Afficher une erreur si la requête n'a pas pu être exécuté
    else
    {
        echo "Un problème est survenu. Veuillez réessayer ultérieurement.";
    }
}

// Fermer la requête
unset($stmt);
}

// Fermer la connexion
unset($pdo);
?>

```

Code pour la suspension d'un compte

```

<?php
// Ouvrir la session, permet d'accéder à la variable &_SESSION
session_start();

// Inclure le fichier config
require_once "../config.php";

// Définition et initialisation des variables
$id = $_POST['id'];
$username = $_POST['username'];
$password = $_POST['password'];
$role = $_POST['role'];

$error = $password_err = "";

if(!empty($username))
{
    // Préparation d'une requête update
    $sql = 'UPDATE utilisateurs SET NomUser = :username WHERE id = :id';
    $stmt = $pdo->prepare($sql);

    // Liaison des variables à la requête en tant que paramètres
    $stmt->bindParam(':username', $username, PDO::PARAM_STR);
    $stmt->bindParam(':id', $id, PDO::PARAM_INT);

    // Exécution de la requête préparée
    if(!$stmt->execute())
    {
        $error = "Un problème est survenu.";
    }

    // Fermer la requête
    unset($stmt);
}

if(!empty($password))
{
    if(strlen(trim($password)) < 8)
    {
        $password_err = "Le mot de passe doit contenir un minimum de 8 caractères.";
    }
    else
    {
        // Préparation d'une requête update
        $sql = 'UPDATE utilisateurs SET MotDePasse = :password WHERE id = :id';
        $stmt = $pdo->prepare($sql);

        // Liaison des variables à la requête en tant que paramètres
        $stmt->bindParam(':password', $password, PDO::PARAM_STR);
        $stmt->bindParam(':id', $id, PDO::PARAM_INT);

        $password = password_hash($password, PASSWORD_DEFAULT); // Crédit à un hachage du mot de passe

        // Exécution de la requête préparée
        if(!$stmt->execute())
        {
            $error = "Un problème est survenu.";
        }

        // Fermer la requête
        unset($stmt);
    }
}

if(!empty($role))
{
    // Préparation d'une requête update
    $sql = 'UPDATE utilisateurs SET Role = :role WHERE id = :id';
    $stmt = $pdo->prepare($sql);

    // Liaison des variables à la requête en tant que paramètres
    $stmt->bindParam(':role', $role, PDO::PARAM_STR);
    $stmt->bindParam(':id', $id, PDO::PARAM_INT);

    // Exécution de la requête préparée
    if(!$stmt->execute())
    {
        $error = "Un problème est survenu.";
    }

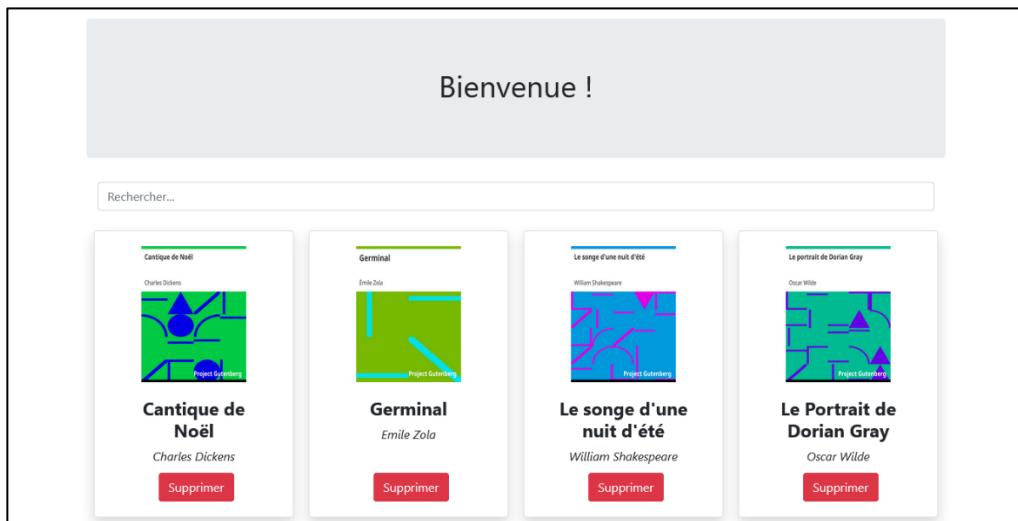
    // Fermer la requête
    unset($stmt);
}

if(empty($error) && empty($password_err))
{
    // Redirection vers page gestion des utilisateurs
    header('Location: GestionUser.php');
    exit();
}
else
{
    echo $error;
    echo $password_err;
}

// Fermer la connexion
unset($pdo);
?>
```

Code pour la modification d'un compte

J'ai aussi ajouté une fonctionnalité qui permet aux comptes "admin" de supprimer des livres du catalogue.



Le catalogue vu par un compte ayant pour rôle "admin"

```
<?php
// Ouvrir la session, permet d'accéder à la variable &_SESSION
session_start();

// Inclure le fichier config
require_once "../config.php";

// Définition et initialisation des variables
$id = $_POST['id'];

// Préparation d'une requête delete
$sql = 'DELETE FROM livres WHERE id = :id';
$stmt = $pdo->prepare($sql);

// Liaison des variables à la requête en tant que paramètres
$stmt->bindParam(':id', $id, PDO::PARAM_INT);

// Exécution de la requête préparée
if ($stmt->execute())
{
    // Redirection vers page gestion des utilisateurs
    header('Location: ../index.php');
    exit();

    // Fermer la requête
    unset($stmt);
}

// Fêmer la connexion
unset($pdo);
?>
```

Le code qui permet de supprimer un livre

Ensuite, j'ai implémenté un site de gestion de compte qui permet aux utilisateurs ayant le rôle "user" de changer leur mot de passe et de visualiser leur nom d'utilisateur ainsi que leur rôle.

Mes informations

Nom d'utilisateur: VotreNom
Veuillez contacter votre administrateur pour changer de nom d'utilisateur.

Rôle: user
Veuillez contacter votre administrateur pour changer de rôle.

Ancien mot de passe: Nouveau mot de passe:

Modifier

Conditions d'Utilisation Politique de Confidentialité

La page de gestion de compte

```
// Traitement des données du formulaire après submit
if($_SERVER['REQUEST_METHOD'] == "POST")
{
    if(!empty($_POST['new-password']))
    {
        // Définition des variables
        $password = $_POST['password'];
        $new_password = $_POST['new-password'];

        // Préparation d'une requête select
        $sql = 'SELECT MotDePasse FROM utilisateurs WHERE id = :id';
        $stmt = $pdo->prepare($sql);

        // Liaison des variables à la requête en tant que paramètres
        $stmt->bindParam(':id', $id, PDO::PARAM_INT);

        // Exécution de la requête préparée
        if($stmt->execute())
        {
            // Récupération du mot de passe de l'utilisateur
            $password_verif = $stmt->fetchColumn();
        }
        else
        {
            echo "Un problème est survenu. Veuillez réessayer ultérieurement.";
        }

        // Vérification de l'ancien mot de passe
        if(empty(trim($password)) == $_SESSION['username'])
        {
            if(strlen(trim($new_password)) < 8)
            {
                $new_password_err = "Le mot de passe doit contenir un minimum de 8 caractères.";
            }
            else
            {
                // Préparation d'une requête update
                $sql1 = "UPDATE utilisateurs SET MotDePasse = :new_password WHERE id = :id";
                $stmt1 = $pdo->prepare($sql1);

                // Liaison des variables à la requête en tant que paramètres
                $stmt1->bindParam(':new_password', $new_password, PDO::PARAM_STR);
                $stmt1->bindParam(':id', $id, PDO::PARAM_INT);

                $new_password = password_hash($new_password, PASSWORD_DEFAULT);

                // Création d'un hashage du mot de passe
                if(empty($error) && empty($password_err) && empty($new_password_error))
                {
                    // Exécution de la requête préparée
                    if($stmt1->execute())
                    {
                        // Confirmation de modification de compte et redirection vers page mon compte
                        echo "<script type='text/javascript'>";
                        alert('Vos informations ont été modifiées.');
                        document.location.href = 'MonCompte.php';
                        </script>";
                    }
                    else
                    {
                        echo "Un problème est survenu. Veuillez réessayer ultérieurement.";
                    }
                }
                else
                {
                    echo "Un problème est survenu. Veuillez réessayer ultérieurement.";
                }

                // Fermer la requête
                unset($stmt1);
            }
        }
        else
        {
            $password_err = "Mot de passe invalide.";
        }
    }
}
```

Le code pour changer son mot de passe

Pour terminer, j'ai mis en place une historique d'emprunt en sauvegardant l'identifiant de l'utilisateur, l'identifiant du livre et la date dans la base de données lors qu'un utilisateur emprunte un livre.

```
// Définition et initialisation des variables
$file = $_POST['file'];
$file_path = str_replace("Bibliotheque-Sterling", "", $file); //Enlever partie du chemin pour redirection
$id_livre = $_POST['id'];
$id_user = $_SESSION['id'];
$date = date("Y-m-d H:i:s");

// Préparation d'une requête delete
$sql = 'INSERT INTO emprunter (NumUser, NumLivre, DateEmprunt) VALUES (:id_user, :id_livre, :date)';
$stmt = $pdo->prepare($sql);

// Liaison des variables à la requête en tant que paramètres
$stmt->bindParam(':id_user', $id_user, PDO::PARAM_INT);
$stmt->bindParam(':id_livre', $id_livre, PDO::PARAM_INT);
$stmt->bindParam(':date', $date, PDO::PARAM_STR);

// Exécution de la requête préparée
if ($stmt->execute())
{
    // Redirection vers le fichier du livre
    header('Location: ' . $file_path . '');
    exit();

    // Fermer la requête
    unset($stmt);
}

// Fêmer la connexion
unset($pdo);
```

Le code pour enregistrer les emprunts

Compte rendu d'Activité		Fiche n° 5
Libellé court	Projet Personnel Encadré E5 (Client lourd)	
Description	Compte rendu d'un projet de développement d'un site de bibliothèque.	
Contexte	Mission accomplie dans le cadre de la formation	

Localisation	<input type="checkbox"/> Entreprise <input checked="" type="checkbox"/> CFA
Source	<input type="checkbox"/> 1ère année <input type="checkbox"/> 2ème année <input type="checkbox"/> TP <input checked="" type="checkbox"/> PPE
Cadre	<input checked="" type="checkbox"/> Seule <input type="checkbox"/> En équipe
Type	<input checked="" type="checkbox"/> Vécu <input type="checkbox"/> Observé <input type="checkbox"/> Simulé

Environnement Technologique	Qt Creator, WAMP, phpMyAdmin
Moyens	C++, Qt, MySQL, GitHub, Krita, FL Studio
Avis personnel	Projet de développement d'une application de bureau qui m'a permis de me familiariser avec Qt et d'approfondir mes compétences en C++.

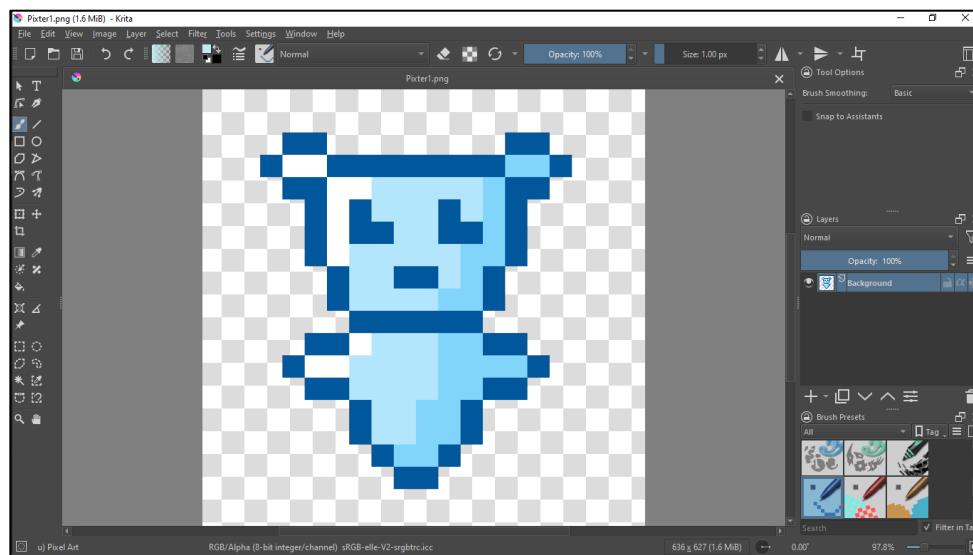
Activités mises en œuvre pour la réalisation de cette situation	
1.6	Organisation de son développement professionnel
B2.1	Conception et développement d'une solution applicative
B2.3	Gestion des données

Déroulement de l'activité :

Dans le contexte de la M2L (Maison des Ligues de Lorraine), j'ai mis en place une application en client lourd, plus précisément une application de bureau, "Pixters", qui permet de s'occuper d'un animal de compagnie virtuel, appelé un "pixter".

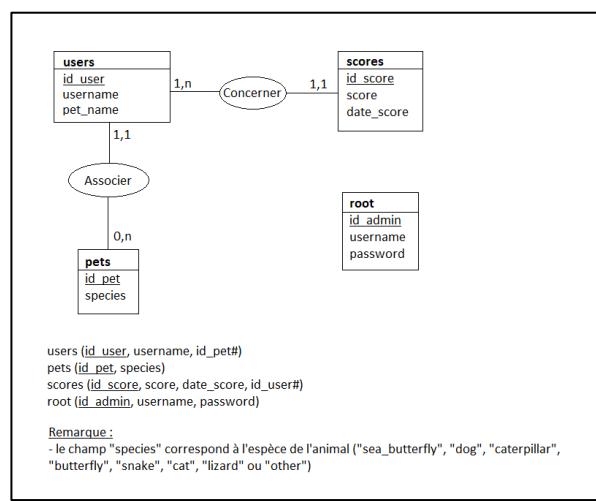
Tout d'abord, j'ai choisi les technologies les mieux adaptés à mes besoins. J'ai décidé d'utiliser C++ avec Qt pour la création d'interfaces graphiques. Ce choix s'est fait sur la base de leurs documentations détaillées et de ma familiarité préalable avec le langage C++. J'ai également opté pour une base de données MySQL. Le projet a été développé dans Qt Creator, l'environnement de développement intégré destiné à Qt.

J'ai commencé par la création des graphismes dans Krita, un logiciel de peinture numérique.



L'interface de Krita

J'ai également réalisé le MCD (Modèle Conceptuel des Données) et le MLD (Modèle Logique des Données) des tables dont j'aurai besoin dans la base de données.

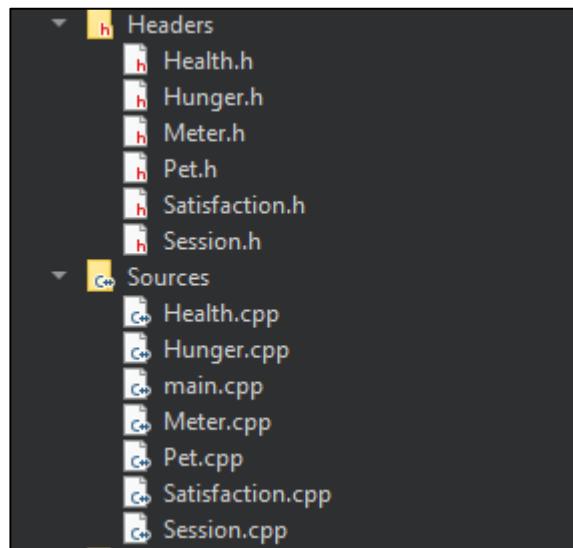


Le MCD et le MLD de la base de données

Ensuite, j'ai mis en place la base de données MySQL dans phpMyAdmin.

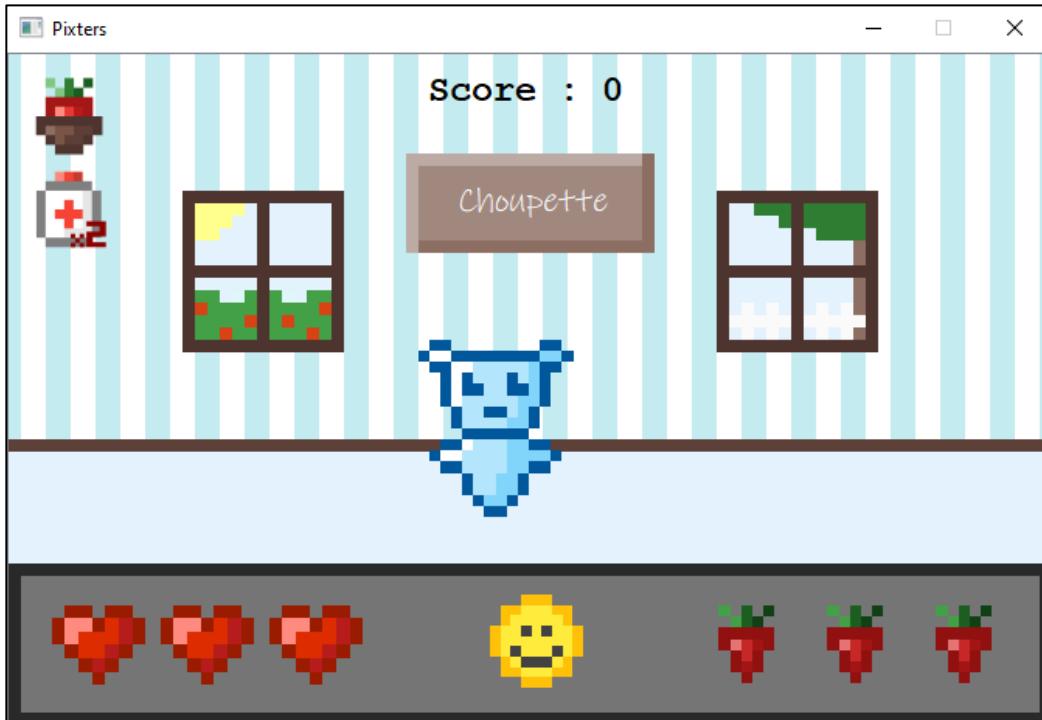
La base de données dans phpMyAdmin

Après avoir créé le projet, je l'ai publié sur GitHub pour la gestion de versions avec Git. Puis, j'ai mis en place les différentes classes dans Qt Creator.



Les classes

Par la suite j'ai ajouté l'arrière-plan et des tableaux d'objets pour afficher la barre de vie et la barre de faim, ainsi qu'une icône qui indique la satisfaction et un panneau qui affiche le nom du pixter.

*L'interface de l'application*

```
//set background
setBackgroundBrush(QBrush(QImage(":/images/Resources/Background.png")));

//create the pet
m_pet = new Pet(m_pet_type);
m_pet->setPos(265,170);
m_pet->setName(m_pet_name);

//create label with name of pet
m_name = new QLabel();
m_name->setText(m_pet->getName());
m_name->setTextFormat(Qt::PlainText);
m_name->setAlignment(Qt::AlignCenter);
m_name->setGeometry(270,80,135,30); //x, y, w, h
m_name->setAttribute(Qt::WA_TranslucentBackground);
m_name->setStyleSheet("font-size: 22px; font-family: 'Ink Free', 'Segoe Print' ; color: white");

//create the meter
m_meter = new Meter(m_scene);

//add pet to scene
m_scene->addItem(m_pet);
m_scene->addWidget(m_name);

//play background music
m_music->setSource(QUrl("qrc:/sounds/Resources/Sounds/background_music.wav"));
m_music->setLoops(QMediaPlayer::Infinite);
m_music->play();

//trigger game over once death value is changed
connect(m_meter->m_morale, SIGNAL(valueChanged()), this, SLOT(gameOver()));
```

Le code de l'interface

J'ai également créé une fonction pour le mouvement du pixter, afin que celui-ci se déplace aléatoirement à droite ou à gauche tout en restant dans un cadre donné. L'image change en fonction de la direction dans laquelle il se déplace.

```

void Pet::move()
{
    QPointF position;
    position = scenePos();
    int posX = position.x();

    //prevent pet going out of bounds
    if(posX <= 20)
    {
        setPos(x()+20, y());
    }
    else if(posX >= 500)
    {
        setPos(x()-20, y());
    }

    //move pet randomly left and right, mirror icon when going right
    if (rand() % 2 + 1 == 1)
    {
        setPos(x()-20, y());
        setPixmap(m_pixter.scaledToHeight(140));
    }
    else
    {
        setPos(x()+20, y());
        setPixmap((m_pixter.scaledToHeight(140)).transformed(QTransform().scale(-1, 1)));
    }
}

```

Le code de la fonction du mouvement

Ensuite, j'ai mis en place le fonctionnement des barres de statut. Pour cela, j'ai connecté un minuteur à la barre de faim qui sert à cacher les icônes au fur et à mesure du passage du temps et changent l'indicateur de satisfaction en conséquence. Si toutes les icônes de la barre de faim sont invisibles, les icônes de la barre de santé diminuent à leur tour de la même façon. Si ces dernières sont elles aussi toutes invisibles, l'indicateur de satisfaction indique la mort du pixter.

```

void Meter::starve()
{
    //reduce number of berries by 1
    if(m_berry[0]->getHunger() == 3)
    {
        m_berry[2]->hungerDown();
        m_berry[0]->setHunger(2);
    }
    else if(m_berry[0]->getHunger() == 2)
    {
        m_berry[1]->hungerDown();
        m_berry[0]->setHunger(1);
    }
    else if(m_berry[0]->getHunger() == 1)
    {
        m_berry[0]->hungerDown();
        m_berry[0]->setHunger(0);
    }
}

```

Le code pour cacher les icônes de la barre de faim

```

void Meter::healthDecrease()
{
    //reduce number of hearts by 1 when berries are depleted
    if(m_berry[0]->getHunger() == 0)
    {
        if(m_heart[0]->getHealth() == 3)
        {
            m_heart[0]->healthDown();
            m_heart[0]->setHealth(2);
        }
        else if(m_heart[0]->getHealth() == 2)
        {
            m_heart[1]->healthDown();
            m_heart[0]->setHealth(1);

            //play alert sound
            QAudioOutput * audioOutput = new QAudioOutput(this);
            QMediaPlayer * sound = new QMediaPlayer(this);
            sound->setAudioOutput(audioOutput);
            audioOutput->setVolume(50);
            sound->setSource(QUrl("qrc:/sounds/Resources/Sounds/alert.wav"));
            sound->play();
        }
        else if(m_heart[0]->getHealth() == 1)
        {
            m_heart[0]->healthDown();
            m_heart[0]->setHealth(0);
        }
    }
}

```

Le code pour cacher les icônes de la barre de vie

Afin d'avoir un moyen de nourrir le pixter, j'ai implementé un bouton qui restaure une icône de la barre de faim.

```

void Meter::eat()
{
    //play eating sound
    QAudioOutput * audioOutput = new QAudioOutput(this);
    QMediaPlayer * sound = new QMediaPlayer(this);
    sound->setAudioOutput(audioOutput);
    audioOutput->setVolume(50);
    sound->setSource(QUrl("qrc:/sounds/Resources/Sounds/eating.wav"));
    sound->play();

    //increase number of berries by 1
    if(m_berry[0]->getHunger() == 0)
    {
        m_berry[0]->hungerUp();
        m_berry[0]->setHunger(1);
        this->happinessChange();
    }
    else if(m_berry[0]->getHunger() == 1)
    {
        m_berry[1]->hungerUp();
        m_berry[0]->setHunger(2);
        this->happinessChange();
    }
    else if(m_berry[0]->getHunger() == 2)
    {
        m_berry[2]->hungerUp();
        m_berry[0]->setHunger(3);
        this->happinessChange();
    }
}

```

La fonction pour nourrir le pixter

Par la suite, j'ai mis en place un écran de fin de jeu déclenché par la mort du pixter (quand l'indicateur de satisfaction atteint le statut "mort") qui contient un bouton pour quitter l'application et un bouton pour commencer une autre partie.



L'écran de fin de jeu

```
//add play again button
m_play_again = new QPushButton();
m_play_again->setIcon(QIcon(":/images/Resources/Play_Again.png"));
m_play_again->setIconSize(QSize(200, 80));
m_play_again->move(380, 90);
m_play_again->setStyleSheet(QString("background-color: rgba(255, 255, 255, 0); color: black;"));
m_play_again->setCursor(Qt::PointingHandCursor);

m_play_again->show();
m_scene->addWidget(m_play_again);

//restart session when play again button is clicked
connect(m_play_again, SIGNAL(clicked()), this, SLOT(restartSession()));

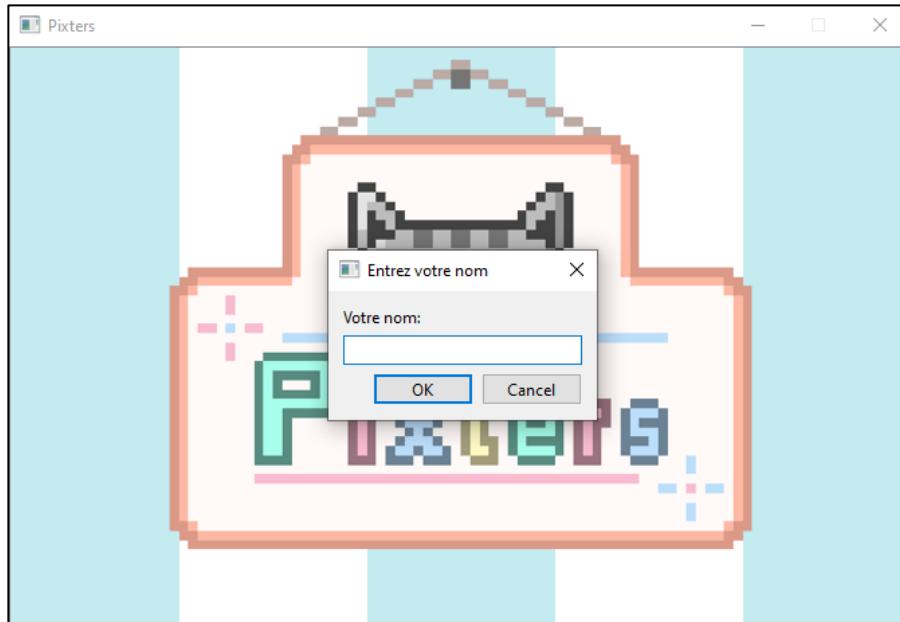
//add quit button
m_quit = new QPushButton();
m_quit->setIcon(QIcon(":/images/Resources/Quit.png"));
m_quit->setIconSize(QSize(200, 80));
m_quit->move(380, 200);
m_quit->setStyleSheet(QString("background-color: rgba(255, 255, 255, 0); color: black;"));
m_quit->setCursor(Qt::PointingHandCursor);

m_quit->show();
m_scene->addWidget(m_quit);

//close program when quit button is clicked
connect(m_quit, SIGNAL(clicked()), this, SLOT(close()));
```

Le code des boutons de l'écran de fin de jeu

Après cela, j'ai créé l'écran titre. Cliquer sur le logo lance une session qui demande d'abord de saisir un nom d'utilisateur, une espèce et un nom pour le pixter avant de démarrer une partie.

*L'écran titre et la demande de saisie*

```
//enter username
bool ok;
m_username = QInputDialog::getText(this, "Entrez votre nom",
                                    "Votre nom:", QLineEdit::Normal, 0, &ok);

//return to title screen on cancel
if(ok == false)
{
    titleScreen();
}
else
{
    //initialize username if no name entered
    if(m_username.length() == 0)
    {
        m_username = "joueur";
    }
}
```

Le code pour la saisie du nom d'utilisateur

Par la suite, j'ai ajouté une base de données pour stocker les informations saisies par l'utilisateur et éventuellement les scores.

```
//connect to database
QSqlDatabase db = QSqlDatabase::addDatabase("QMYSQL");
db.setHostName("localhost");
db.setDatabaseName("pixters");
db.setUserName("root");
db.setPassword("Sio2022!"); //phpMyAdmin password

//open database
if(!db.open())
{
    qDebug() << "DB Connect Error: " << db.lastError();
}
```

La connexion à la base de données

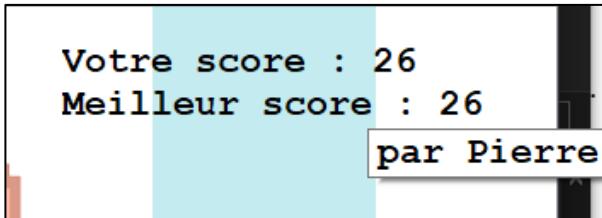
```
//prepare insert statement
QSqlQuery insert_val;
insert_val.prepare("INSERT INTO users (username, pet_name, id_pet)"
    "VALUES (:username, :pet_name, :id_pet)");
insert_val.bindValue(":username", m_username);
insert_val.bindValue(":pet_name", m_pet_name);
insert_val.bindValue(":id_pet", pet_type);

//execute statement
if(!insert_val.exec())
{
    qDebug() << "SQL Statement Error: " << insert_val.lastError();
}
```

L'ajout des informations dans la base de données

De plus, j'ai changé le panneau de l'écran de jeu pour qu'il affiche le nom du pixter saisi par l'utilisateur. L'image du pixter dépend également de l'espèce choisie auparavant. Afin que le nom ne soit pas trop long pour le panneau, j'ai limité le nombre de lettres qu'on peut saisir.

J'ai par la suite ajouté un affichage pour le score actuel de l'utilisateur ainsi que pour le meilleur score sur l'écran de fin de jeu. En passant la souris au-dessus du meilleur score, on peut visualiser le nom de l'utilisateur qui y est associé.



L'affichage du score et du meilleur score sur l'écran de fin de jeu

```
//store score in variable
int score;
score = m_meter->getScore();

//prepare select statement for high score
QSqlQuery get_highscore;
get_highscore.exec("SELECT score, username FROM scores, users WHERE scores.id_user=users.id_user AND score=(SELECT max(score) FROM scores)");

//execute statement
if(!get_highscore.exec())
{
    qDebug() << "SQL Statement Error: " << get_highscore.lastError();
}

//store result in variable (@ = column index)
while (get_highscore.next())
{
    m_highscore = get_highscore.value(0).toInt();
    m_highscore_user = get_highscore.value(1).toString();
}

//set new high score if session score higher than high score
if(m_highscore < score)
{
    m_highscore = score;
    m_highscore_user = m_username;
}

//prepare update statement for high score
QSqlQuery insert_score;
insert_score.prepare("INSERT INTO scores(score, date, id_user)"
    "VALUES (:score, :date, :id_user)");
insert_score.bindValue(":score", score);
insert_score.bindValue(":date", QDateTime::currentDateTime().toString());
insert_score.bindValue(":id_user", m_id_user);

//execute statement
if(!insert_score.exec())
{
    qDebug() << "SQL Statement Error: " << insert_score.lastError();
}
```

Le code pour l'affichage du score sur l'écran de fin de jeu

Ensuite, j'ai ajouté des mentions légales sur l'écran titre.

```
void Session::showContract()
{
    QMessageBox msgBox;
    msgBox.setText("Pixters est édité par Laura Huber.\n"
                  "Tous les éléments de ce site, y compris les graphismes et les bruitages, ont été créés sur mesure. La reproduction des éléments "
                  "de cette application est autorisée à la condition d'y mentionner la source. Elles ne peuvent être utilisées à des fins commerciales "
                  "et publicitaires.");
    msgBox.exec();
}
```

Les mentions légales

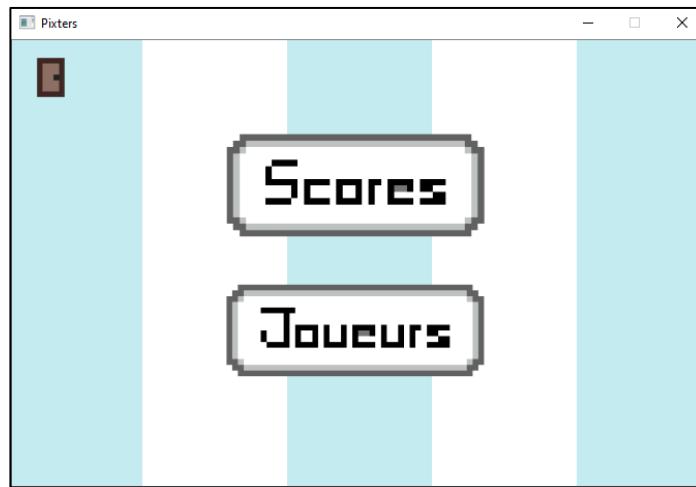
Afin de pouvoir soigner le pixter, j'ai mis en place une fonctionnalité de médecine sous la forme d'un bouton qui restaure un cœur. Le nombre de médicaments est limité pour que le niveau de difficulté ne soit pas trop bas et augmente de 1 toutes les quelques secondes, jusqu'à un maximum de trois médicaments.

```
void Meter::healthIncrease()
{
    //play healing sound
    QAudioOutput * audioOutput = new QAudioOutput(this);
    QMediaPlayer * sound = new QMediaPlayer(this);
    sound->setAudioOutput(audioOutput);
    audioOutput->setVolume(50);
    //if medicine not empty play healing sound
    if(m_medicine > 0)
    {
        sound->setSource(QUrl("qrc:/sounds/Resources/Sounds/bweep.wav"));
    }
    //if medicine empty play alert sound
    else
    {
        sound->setSource(QUrl("qrc:/sounds/Resources/Sounds/no_medicine.wav"));
    }
    sound->play();

    //do not use medicine if health is full
    if(m_heart[0]->getHealth() < 3)
    {
        //only heal when medicine not empty
        if(m_medicine != 0)
        {
            //increase number of hearts by 1
            if(m_heart[0]->getHealth() == 0)
            {
                m_heart[0]->healthUp();
                m_heart[0]->setHealth(1);
            }
            else if(m_heart[0]->getHealth() == 1)
            {
                m_heart[1]->healthUp();
                m_heart[0]->setHealth(2);
            }
            else if(m_heart[0]->getHealth() == 2)
            {
                m_heart[2]->healthUp();
                m_heart[0]->setHealth(3);
            }
        }
        //change icon according to number of medicines and deplete medicine
        if(m_medicine == 2)
        {
            m_heal->setIcon(QIcon(":/images/Resources/Medicine1.png"));
            m_medicine--;
        }
        else if(m_medicine == 1)
        {
            m_heal->setIcon(QIcon(":/images/Resources/Medicine0.png"));
            m_medicine--;
        }
    }
}
```

Le code pour la fonctionnalité de médecine

À la fin, j'ai ajouté un compte d'administrateur qui peut accéder à un menu d'administration qui permet, après connexion, de visualiser une liste des utilisateurs et les scores.



Le menu d'administration

```

if(password == m_password && username == m_root)
{
    //add back button
    m_back = new QPushButton();
    m_back->setIcon(QIcon(":/images/Resources/Door.png"));
    m_back->setIconSize(QSize(38, 38));
    m_back->move(5, -18);
    m_back->setToolTip("Quitter");
    m_back->setStyleSheet(QString("background-color: rgba(255, 255, 255, 0); color: black;"));
    m_back->setCursor(Qt::PointingHandCursor);

    m_back->show();
    m_scene->addWidget(m_back);

    //show scoreboard when scoreboard button is clicked
    connect(m_back, SIGNAL(clicked()), this, SLOT(titleScreen()));

    //add scoreboard button
    m_scoreboard = new QPushButton();
    m_scoreboard->setIcon(QIcon(":/images/Resources/Scoreboard.png"));
    m_scoreboard->setIconSize(QSize(250, 130));
    m_scoreboard->move(200, 40);
    m_scoreboard->setStyleSheet(QString("background-color: rgba(255, 255, 255, 0); color: black;"));
    m_scoreboard->setCursor(Qt::PointingHandCursor);

    m_scoreboard->show();
    m_scene->addWidget(m_scoreboard);

    //show scoreboard when scoreboard button is clicked
    connect(m_scoreboard, SIGNAL(clicked()), this, SLOT(showScoreboard()));

    //add player list button
    m_player_list = new QPushButton();
    m_player_list->setIcon(QIcon(":/images/Resources/Players.png"));
    m_player_list->setIconSize(QSize(250, 130));
    m_player_list->move(200, 180);
    m_player_list->setStyleSheet(QString("background-color: rgba(255, 255, 255, 0); color: black;"));
    m_player_list->setCursor(Qt::PointingHandCursor);

    m_player_list->show();
    m_scene->addWidget(m_player_list);

    //show player list when player list button is clicked
    connect(m_player_list, SIGNAL(clicked()), this, SLOT(showUsers()));
}

else
{
    QMessageBox msgBox;
    msgBox.setText("Nom d'utilisateur ou mot de passe incorrect.");
}

```

Le code du menu d'administration

Pour terminer, j'ai ajouté des musiques de fond ainsi que des bruitages et des info-bulles quand on passe la souris au-dessus des boutons.