Rapport de stage projet JVSPJ



https://github.com/Gotlub/OldFacture

Introduction:

En cours d'année 2022, la communauté de communes du Pays de l'Aigle a procédé au changement de son logiciel de gestion, passant du logiciel JVS à Ciril. Au cours de cette transition, il est apparu que certaines fonctionnalités n'étaient plus compatibles, notamment la recherche et le téléchargement des anciennes factures. Le projet JVS PJ (PJ pour pièce jointe) vise donc à permettre le téléchargement des anciennes factures.

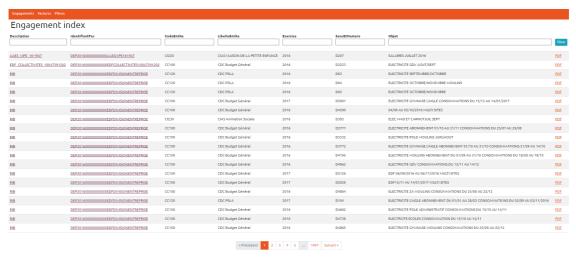
Outils pour la réalisation du projet :

- Python pour gérer le routage des chemins des PDF
- MySQL pour la gestion de la base de données
- Symfony 6 et PHP version supérieure à 8

Mise en place du projet :

Il existe 170 000 PDF dont les noms sont répertoriés dans la BDD, mais dont les chemins d'accès ne sont plus valides. Après l'ajout d'une nouvelle colonne à la base de données et l'utilisation d'un script Python pour localiser les fichiers, les PDF sont désormais correctement répertoriés.

Dans la base de données, trois tables distinctes (factures, pièces, engagements) sont indépendantes les unes des autres et ne sont pas interconnectées.



En raison du très grand nombre de fichiers, une pagination a été configurée.

Téléchargement des fichiers:

Afin de permettre l'envoi de certains champs de recherche par lien ou leur sauvegarde, les filtres de recherche sont transmis via la méthode « GET » dans l'URL pour les trois pages concernées.

La vue:

```
{% block body %}
  {% if descrip|default(null) %}{% endif %}
  {% if ident|default(null) %}{% endif %}
{% if code|default(null) %}{% endif %}
{% if code|default(null) %}{% endif %}
{% if libelle|default(null) %}{% endif %}
{% if annee|default(null) %}{% endif %}
{% if obj|default(null) %}{% endif %}
{% if obj|default(null) %}{% endif %}
  div class="container-fluid">
    cdiv class-"row
       <h1>Facture index</h1>
       cform class="form-inline mt-1" method="POST" action="{{ path('app_facture.findallcontain') }}">
                  cth class="text-left align-top" scope="col">Description
                    IdentifiantPes
                    CodeEntite
                    LibelleEntite
                    Exercice
                     <div class="form-group mr+1 mb-2">
                       <input type="number" class="input-group input-group-lg" name="Exercice" min = "2015"
   value="{% if annee|default and annee != "**" %}{{ annee }}{% endif %}">
                  Numero
                       <input type="number" class="input-group input-group-lg" name="Numero"
    value="{% if numero|default and numero != "**" %}{{ numero }}{% endif %}">
                  Objet
```

Les contrôleurs :

```
$code - ($code -= "") ? "" : $code;
$obj - $request->get("Objet");
$obj - ($obj -= "") ? "" : $cobj;
$libelle - $request->get("LibelleEntite");
$libelle - ($libelle -= "") ? """ : $libelle;
$ident - $request->get("identifiantPes");
$ident - ($ident -= "") ? """ : $ident;
$request ->get("Moreon");
     $numero = $request->get("Numero");
$numero = ($numero == "") ? *** : $numero;
     return $this->redirectToRoute('app_facture.findallcontainReq', [
          'annee' -> $annee,
'descrip' -> $descrip,
'ident' -> $ident,
'code' -> $code,
          'obj' -> $obj,
'libelle' -> $libelle,
'numero' -> $numero
if($descrip != "++")(
    $params += array("description" -> $descrip);
     if($code != "**"){
          $params += array("code_entite" -> $code);
     if($obj != "**"){
          $params += array("objet" -> $obj);
     if($libelle !- "**"){
          $params +- array("libelle_entite" -> $libelle);
     if($ident != "**"){
          $params += array("identifiant_PES" -> $ident);
     if($numero !- "***){
          $params +- array("numero" -> $numero);
     if(count($params) -- 0) {
          return $this->redirectToRoute('app_facture.index');
     $pagination = $paginator->paginate(
          $factureRepository->paginationQueryComplex($params)->getResult(),
          $request->query->get('page', 1),
     return $this->render('facture/index.html.twig', [
    'pagination' -> $pagination,
          'descrip' -> $descrip,
'code' -> $code,
          'ident' -> $ident,
           'annee' -> $annee,
          'libelle' -> $libelle,
'numero' -> $numero
```

Le repository:

Les requêtes pour effectuer les recherches sont effectué en SQL puis converti en DQL (Doctrine Query Langage, l'ORM de Symfony) pour être compatible avec le bundle qui sert à la pagination (knp_paginator).



Le site étant très simple il a été terminé et déployé sur le réseau local en trois jours.

