Projet d’indexation de PDF

Objectif

L'objectif est de construire une application capable de prendre en entrée des fichier PDF et de permettre à un utilisateur de faire rechercher dans l’ensemble de ces documents.

Dans un premier temps, l'interface d'administration automatisée de Django permettra l’ajout des documents et leur traitement. La partie visible du logiciel ne permettra que la recherche.

Ce traitement consiste à transformer le fichier PDF en autant d'images que de pages, puis, dans un second temps, à extraire le texte de chaque image, grâce à de l'OCR. Le code technique sur cette partie est déjà réalisé.

Montée en compétence nécessaire

- Tutoriel Django : https://docs.djangoproject.com/en/5.0/ref/contrib/gis/tutorial/
- Tutoriel VueJS : https://fr.vuejs.org/tutorial/#step-1
- InertiaJS pour Django : https://github.com/inertiajs/inertia-django
- VueJS : https://github.com/mujahidfa/inertia-django-vite-vue-minimal
- full text + trigramme : https://docs.djangoproject.com/en/5.0/ref/contrib/postgres/search/
- Celery : https://docs.celeryq.dev/en/stable/django/first-steps-with-django.html

Travail à effectuer

Modèle :

- DocumentPdf
  - Titre (obligatoire, déduit du nom du fichier, modifiagle)
  - Date du document (obligatoire, saisie à la main)
  - Fichier original (obligatoire, uploadé à la création)
  - Slug (obligatoire, déduit du nom du fichier)
  - Nombre de page (obligatoire, déduit du fichier)
- Page
  - FK vers un Document
  - Numéro de page (auto-incrémentée en commençant à 1 et dépendant du document)
  - Image (obligatoire)
  - Texte (non obligatoire)

Interface d'administration

- Possibilité de voir la liste des documents, filtrés par mot clé
- Possibilité de voir la liste des pages, filtrées par document
- Possibilité d'ajouter un document
- Ajouter la création des pages à partir du document (code technique déjà écrit)
- Ajouter la transformation de l'image d'une page en texte (code technique déjà écrit)

Commandes

- Pour la création des pages à partir du document (même code à réutiliser)
- Pour la transformation manuelle de l'image d'une page en texte (même code à réutiliser)

Programmation distribuée

Rendre asynchrone les deux tâches précédentes à l'aide de Celery (ou d'une alternative)

Vue

- Page de recherche full-text :
  - un formulaire permettant de saisie un texte.
  - Le résultat dans un tableau (titre document, date document, numéro page, extrait)
  - Gérer la pagination le résultat, filtrer par titre de document, par date, mais coté Vue
- Page de recherche par trigramme :
  - Pareil, mais en utilisant la recherche par trigramme
- Page d'accueil avec un lien vers chacune des deux pages précédentes
- Page Mention légales avec les informations sur le produit développé

Bonus

- Page permettant l'upload d'un fichier PDF directement
- Page permettant l'upload de plusieurs fichiers PDF (on cible un répertoire). Vue va alors proposer de rentrer le titre et la date du document dans un tableau, en face de chaque fichiers, puis va les uploader un à un)
- Possibilité d'intégrer une bibliothèque CSS un peu sympa (TailWind ?)
- Possibilité de gérer l'upload d'une archive ZIP (ou autre format d’archive) et de gérer le titre et la date des documents dans un second temps
- Possibilité d’enregistrer une recherche en liant une citation à la page correspondante
- Travail spécifique pour organiser le texte brut en sections (liées à une ou plusieurs pages)
- Travail spécifique pour gérer les tableaux (liés à une ou plusieurs pages)