Dossier du personnel (dossiers individuels)". Cette fonctionnalité permettra de gérer les documents liés à chaque employé, tels que le CV, les diplômes, les certificats de travail, etc.

Voici le code détaillé :

1. Modèle (ajoutez ceci à models.py) :

from django.db import models

from employe.models import Employe

class DocumentPersonnel(models.Model):

TYPE\_CHOICES = [

('CV', 'Curriculum Vitae'),

('DIPLOME', 'Diplôme'),

('CERTIFICAT', 'Certificat de travail'),

('FORMATION', 'Attestation de formation'),

('AUTRE', 'Autre document'),

]

employe = models.ForeignKey(Employe, on\_delete=models.CASCADE, related\_name='documents')

type\_document = models.CharField(max\_length=20, choices=TYPE\_CHOICES)

titre = models.CharField(max\_length=255)

fichier = models.FileField(upload\_to='documents\_personnels/')

date\_ajout = models.DateField(auto\_now\_add=True)

commentaire = models.TextField(blank=True, null=True)

def \_\_str\_\_(self):

return f"{self.get\_type\_document\_display()} - {self.employe.nom} {self.employe.prenoms}"

1. Formulaire (ajoutez ceci à forms.py) :

from django import forms

from .models import DocumentPersonnel

class DocumentPersonnelForm(forms.ModelForm):

class Meta:

model = DocumentPersonnel

fields = ['employe', 'type\_document', 'titre', 'fichier', 'commentaire']

1. Vues (ajoutez ceci à views.py) :

from django.shortcuts import render, redirect, get\_object\_or\_404

from django.contrib import messages

from employe.models import Employe

from .models import DocumentPersonnel

from .forms import DocumentPersonnelForm

def dossier\_personnel(request):

employes = Employe.objects.all()

return render(request, 'gpec/dossier\_personnel.html', {'employes': employes})

def detail\_dossier(request, employe\_id):

employe = get\_object\_or\_404(Employe, id=employe\_id)

documents = employe.documents.all().order\_by('-date\_ajout')

return render(request, 'gpec/detail\_dossier.html', {'employe': employe, 'documents': documents})

def ajouter\_document(request, employe\_id):

employe = get\_object\_or\_404(Employe, id=employe\_id)

if request.method == 'POST':

form = DocumentPersonnelForm(request.POST, request.FILES)

if form.is\_valid():

document = form.save(commit=False)

document.employe = employe

document.save()

messages.success(request, "Document ajouté avec succès.")

return redirect('detail\_dossier', employe\_id=employe.id)

else:

form = DocumentPersonnelForm(initial={'employe': employe})

return render(request, 'gpec/ajouter\_document.html', {'form': form, 'employe': employe})

def supprimer\_document(request, document\_id):

document = get\_object\_or\_404(DocumentPersonnel, id=document\_id)

employe\_id = document.employe.id

if request.method == 'POST':

document.delete()

messages.success(request, "Document supprimé avec succès.")

return redirect('detail\_dossier', employe\_id=employe\_id)

return render(request, 'gpec/confirmer\_suppression.html', {'document': document})

1. URLs (ajoutez ceci à urls.py) :

from django.urls import path

from . import views

urlpatterns = [

# ... autres URLs ...

path('gpec/dossier-personnel/', views.dossier\_personnel, name='dossier\_personnel'),

path('gpec/detail-dossier/<int:employe\_id>/', views.detail\_dossier, name='detail\_dossier'),

path('gpec/ajouter-document/<int:employe\_id>/', views.ajouter\_document, name='ajouter\_document'),

path('gpec/supprimer-document/<int:document\_id>/', views.supprimer\_document, name='supprimer\_document'),

]

1. Templates :
   1. dossier\_personnel.html :

{% extends 'base.html' %}

{% block content %}

<h1>Dossiers du Personnel</h1>

<table class="table">

<thead>

<tr>

<th>Employé</th>

<th>Nombre de documents</th>

<th>Actions</th>

</tr>

</thead>

<tbody>

{% for employe in employes %}

<tr>

<td>{{ employe.nom }} {{ employe.prenoms }}</td>

<td>{{ employe.documents.count }}</td>

<td>

<a href="{% url 'detail\_dossier' employe.id %}" class="btn btn-info btn-sm">Voir le dossier</a>

<a href="{% url 'ajouter\_document' employe.id %}" class="btn btn-success btn-sm">Ajouter un document</a>

</td>

</tr>

{% endfor %}

</tbody>

</table>

{% endblock %}

* 1. detail\_dossier.html :

{% extends 'base.html' %}

{% block content %}

<h1>Dossier de {{ employe.nom }} {{ employe.prenoms }}</h1>

<a href="{% url 'ajouter\_document' employe.id %}" class="btn btn-primary mb-3">Ajouter un document</a>

<table class="table">

<thead>

<tr>

<th>Type de document</th>

<th>Titre</th>

<th>Date d'ajout</th>

<th>Actions</th>

</tr>

</thead>

<tbody>

{% for document in documents %}

<tr>

<td>{{ document.get\_type\_document\_display }}</td>

<td>{{ document.titre }}</td>

<td>{{ document.date\_ajout }}</td>

<td>

<a href="{{ document.fichier.url }}" class="btn btn-info btn-sm" target="\_blank">Voir</a>

<a href="{% url 'supprimer\_document' document.id %}" class="btn btn-danger btn-sm">Supprimer</a>

</td>

</tr>

{% endfor %}

</tbody>

</table>

<a href="{% url 'dossier\_personnel' %}" class="btn btn-secondary">Retour à la liste</a>

{% endblock %}

* 1. ajouter\_document.html :

{% extends 'base.html' %}

{% block content %}

<h1>Ajouter un document pour {{ employe.nom }} {{ employe.prenoms }}</h1>

<form method="post" enctype="multipart/form-data">

{% csrf\_token %}

{{ form.as\_p }}

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

</form>

<a href="{% url 'detail\_dossier' employe.id %}" class="btn btn-secondary mt-3">Retour au dossier</a>

{% endblock %}

* 1. confirmer\_suppression.html :

{% extends 'base.html' %}

{% block content %}

<h1>Confirmer la suppression</h1>

<p>Êtes-vous sûr de vouloir supprimer le document "{{ document.titre }}" de {{ document.employe.nom }} {{ document.employe.prenoms }} ?</p>

<form method="post">

{% csrf\_token %}

<button type="submit" class="btn btn-danger">Confirmer la suppression</button>

<a href="{% url 'detail\_dossier' document.employe.id %}" class="btn btn-secondary">Annuler</a>

</form>

{% endblock %}

Ces fichiers fournissent une implémentation complète de la fonctionnalité de gestion des dossiers du personnel. Vous pouvez maintenant :

1. Voir une liste de tous les employés avec le nombre de documents dans leur dossier
2. Voir les détails du dossier d'un employé spécifique
3. Ajouter un nouveau document à un dossier d'employé
4. Supprimer un document existant

N'oubliez pas d'effectuer les migrations de la base de données après avoir ajouté ce nouveau modèle :

python manage.py makemigrations

python manage.py migrate

Assurez-vous également que votre fichier settings.py est configuré correctement pour gérer les fichiers uploadés :

MEDIA\_URL = '/media/'

MEDIA\_ROOT = os.path.join(BASE\_DIR, 'media')

Et ajoutez ceci à votre fichier urls.py principal :

from django.conf import settings

from django.conf.urls.static import static

if settings.DEBUG:

urlpatterns += static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)

Cette implémentation vous donne une base solide pour la gestion des dossiers du personnel dans votre système GPEC.