Le référentiel de compétences". Cette fonctionnalité permettra de créer et gérer un référentiel de compétences pour l'organisation, ainsi que d'évaluer les compétences des employés.

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

from django.db import models

from employe.models import Employe

class Competence(models.Model):

nom = models.CharField(max\_length=200)

description = models.TextField()

categorie = models.CharField(max\_length=100)

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

return self.nom

class NiveauCompetence(models.Model):

NIVEAUX = [

(1, 'Débutant'),

(2, 'Intermédiaire'),

(3, 'Avancé'),

(4, 'Expert'),

]

competence = models.ForeignKey(Competence, on\_delete=models.CASCADE)

employe = models.ForeignKey(Employe, on\_delete=models.CASCADE)

niveau = models.IntegerField(choices=NIVEAUX)

date\_evaluation = models.DateField(auto\_now=True)

class Meta:

unique\_together = ('competence', 'employe')

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

return f"{self.employe.nom} - {self.competence.nom} - Niveau {self.get\_niveau\_display()}"

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

from django import forms

from .models import Competence, NiveauCompetence

class CompetenceForm(forms.ModelForm):

class Meta:

model = Competence

fields = ['nom', 'description', 'categorie']

class NiveauCompetenceForm(forms.ModelForm):

class Meta:

model = NiveauCompetence

fields = ['competence', 'employe', 'niveau']

1. Vues (ajoutez ceci à views.py) :

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

from django.contrib import messages

from .models import Competence, NiveauCompetence, Employe

from .forms import CompetenceForm, NiveauCompetenceForm

def liste\_competences(request):

competences = Competence.objects.all().order\_by('categorie', 'nom')

return render(request, 'gpec/liste\_competences.html', {'competences': competences})

def ajouter\_competence(request):

if request.method == 'POST':

form = CompetenceForm(request.POST)

if form.is\_valid():

form.save()

messages.success(request, "Compétence ajoutée avec succès.")

return redirect('liste\_competences')

else:

form = CompetenceForm()

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

def modifier\_competence(request, competence\_id):

competence = get\_object\_or\_404(Competence, id=competence\_id)

if request.method == 'POST':

form = CompetenceForm(request.POST, instance=competence)

if form.is\_valid():

form.save()

messages.success(request, "Compétence modifiée avec succès.")

return redirect('liste\_competences')

else:

form = CompetenceForm(instance=competence)

return render(request, 'gpec/modifier\_competence.html', {'form': form, 'competence': competence})

def evaluer\_competence(request):

if request.method == 'POST':

form = NiveauCompetenceForm(request.POST)

if form.is\_valid():

form.save()

messages.success(request, "Évaluation de compétence enregistrée avec succès.")

return redirect('liste\_competences')

else:

form = NiveauCompetenceForm()

return render(request, 'gpec/evaluer\_competence.html', {'form': form})

def profil\_competences\_employe(request, employe\_id):

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

niveaux\_competences = NiveauCompetence.objects.filter(employe=employe).order\_by('competence\_\_categorie', 'competence\_\_nom')

return render(request, 'gpec/profil\_competences\_employe.html', {'employe': employe, 'niveaux\_competences': niveaux\_competences})

def matrice\_competences(request):

competences = Competence.objects.all().order\_by('categorie', 'nom')

employes = Employe.objects.all().order\_by('nom')

matrice = {}

for employe in employes:

matrice[employe] = {}

for competence in competences:

try:

niveau = NiveauCompetence.objects.get(employe=employe, competence=competence)

matrice[employe][competence] = niveau.niveau

except NiveauCompetence.DoesNotExist:

matrice[employe][competence] = 0

return render(request, 'gpec/matrice\_competences.html', {'competences': competences, 'employes': employes, 'matrice': matrice})

1. URLs (ajoutez ceci à urls.py) :

from django.urls import path

from . import views

urlpatterns = [

# ... autres URLs ...

path('gpec/competences/', views.liste\_competences, name='liste\_competences'),

path('gpec/ajouter-competence/', views.ajouter\_competence, name='ajouter\_competence'),

path('gpec/modifier-competence/<int:competence\_id>/', views.modifier\_competence, name='modifier\_competence'),

path('gpec/evaluer-competence/', views.evaluer\_competence, name='evaluer\_competence'),

path('gpec/profil-competences/<int:employe\_id>/', views.profil\_competences\_employe, name='profil\_competences\_employe'),

path('gpec/matrice-competences/', views.matrice\_competences, name='matrice\_competences'),

]

1. Templates :
   1. liste\_competences.html :

{% extends 'base.html' %}

{% block content %}

<h1>Référentiel de Compétences</h1>

<a href="{% url 'ajouter\_competence' %}" class="btn btn-primary mb-3">Ajouter une compétence</a>

<a href="{% url 'evaluer\_competence' %}" class="btn btn-secondary mb-3">Évaluer une compétence</a>

<a href="{% url 'matrice\_competences' %}" class="btn btn-info mb-3">Voir la matrice des compétences</a>

<table class="table">

<thead>

<tr>

<th>Catégorie</th>

<th>Nom</th>

<th>Description</th>

<th>Actions</th>

</tr>

</thead>

<tbody>

{% for competence in competences %}

<tr>

<td>{{ competence.categorie }}</td>

<td>{{ competence.nom }}</td>

<td>{{ competence.description|truncatewords:20 }}</td>

<td>

<a href="{% url 'modifier\_competence' competence.id %}" class="btn btn-warning btn-sm">Modifier</a>

</td>

</tr>

{% empty %}

<tr>

<td colspan="4">Aucune compétence trouvée.</td>

</tr>

{% endfor %}

</tbody>

</table>

{% endblock %}

* 1. ajouter\_competence.html :

{% extends 'base.html' %}

{% block content %}

<h1>Ajouter une Compétence</h1>

<form method="post">

{% csrf\_token %}

{{ form.as\_p }}

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

</form>

{% endblock %}

* 1. modifier\_competence.html :

{% extends 'base.html' %}

{% block content %}

<h1>Modifier la Compétence : {{ competence.nom }}</h1>

<form method="post">

{% csrf\_token %}

{{ form.as\_p }}

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

</form>

{% endblock %}

* 1. evaluer\_competence.html :

{% extends 'base.html' %}

{% block content %}

<h1>Évaluer une Compétence</h1>

<form method="post">

{% csrf\_token %}

{{ form.as\_p }}

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

</form>

{% endblock %}

* 1. profil\_competences\_employe.html :

{% extends 'base.html' %}

{% block content %}

<h1>Profil de Compétences - {{ employe.nom }} {{ employe.prenom }}</h1>

<table class="table">

<thead>

<tr>

<th>Catégorie</th>

<th>Compétence</th>

<th>Niveau</th>

</tr>

</thead>

<tbody>

{% for niveau\_competence in niveaux\_competences %}

<tr>

<td>{{ niveau\_competence.competence.categorie }}</td>

<td>{{ niveau\_competence.competence.nom }}</td>

<td>{{ niveau\_competence.get\_niveau\_display }}</td>

</tr>

{% empty %}

<tr>

<td colspan="3">Aucune compétence évaluée pour cet employé.</td>

</tr>

{% endfor %}

</tbody>

</table>

{% endblock %}

* 1. matrice\_competences.html :

{% extends 'base.html' %}

{% block content %}

<h1>Matrice des Compétences</h1>

<div class="table-responsive">

<table class="table table-bordered">

<thead>

<tr>

<th>Employé</th>

{% for competence in competences %}

<th>{{ competence.nom }}</th>

{% endfor %}

</tr>

</thead>

<tbody>

{% for employe, competences\_employe in matrice.items %}

<tr>

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

{% for competence in competences %}

<td>

{% if competences\_employe|get\_item:competence %}

{{ competences\_employe|get\_item:competence }}

{% else %}

-

{% endif %}

</td>

{% endfor %}

</tr>

{% endfor %}

</tbody>

</table>

</div>

{% endblock %}

N'oubliez pas d'effectuer les migrations après avoir ajouté ces nouveaux modèles :

python manage.py makemigrations

python manage.py migrate

Cette implémentation vous permet de :

1. Créer et gérer un référentiel de compétences
2. Évaluer les compétences des employés
3. Voir le profil de compétences d'un employé spécifique
4. Afficher une matrice des compétences pour tous les employés

Assurez-vous d'ajouter un lien vers cette nouvelle fonctionnalité dans votre dashboard GPEC ou dans le menu de navigation principal.

Cette fonctionnalité vous donnera un outil complet pour gérer les compétences au sein de votre organisation, ce qui est essentiel pour le développement des employés, la planification de la formation, et l'identification des besoins en recrutement.