L'intégration de la fonctionnalité de contrôle budgétaire est une excellente façon d'améliorer votre logiciel. Voici les étapes pour l'intégrer à votre système existant :

Étape 1 : Mise à jour des modèles

1. Ouvrez votre fichier models.py et ajoutez les nouveaux modèles :

from django.db import models

from employe.models import Employe # Assurez-vous que ce chemin d'importation est correct

class BudgetSalarial(models.Model):

annee = models.IntegerField(unique=True)

budget\_total = models.DecimalField(max\_digits=12, decimal\_places=2)

budget\_utilise = models.DecimalField(max\_digits=12, decimal\_places=2, default=0)

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

return f"Budget salarial {self.annee}"

def budget\_restant(self):

return self.budget\_total - self.budget\_utilise

class BudgetDepartement(models.Model):

budget = models.ForeignKey(BudgetSalarial, on\_delete=models.CASCADE, related\_name='budgets\_departements')

departement = models.ForeignKey('Structure', on\_delete=models.CASCADE) # Utilisez 'Structure' si c'est votre modèle pour les départements

montant\_alloue = models.DecimalField(max\_digits=12, decimal\_places=2)

montant\_utilise = models.DecimalField(max\_digits=12, decimal\_places=2, default=0)

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

return f"Budget {self.departement.denomination} - {self.budget.annee}"

def montant\_restant(self):

return self.montant\_alloue - self.montant\_utilise

1. Créez et appliquez les migrations :

python manage.py makemigrations

python manage.py migrate

Étape 2 : Mise à jour des vues

1. Ouvrez votre fichier views.py et ajoutez la nouvelle vue :

from django.shortcuts import render

from django.db.models import Sum

from datetime import datetime

from .models import BudgetSalarial, BudgetDepartement, Paiement

def controle\_budgetaire(request):

annee\_courante = datetime.now().year

budget, created = BudgetSalarial.objects.get\_or\_create(annee=annee\_courante, defaults={'budget\_total': 0})

budgets\_departements = BudgetDepartement.objects.filter(budget=budget)

masse\_salariale\_reelle = Paiement.objects.filter(date\_paiement\_\_year=annee\_courante).aggregate(Sum('salaire\_brut'))['salaire\_brut\_\_sum'] or 0

ecart = budget.budget\_total - masse\_salariale\_reelle

context = {

'budget': budget,

'budgets\_departements': budgets\_departements,

'masse\_salariale\_reelle': masse\_salariale\_reelle,

'ecart': ecart,

}

return render(request, 'gpec/controle\_budgetaire.html', context)

Étape 3 : Création du template

1. Créez un nouveau fichier controle\_budgetaire.html dans votre dossier de templates (probablement templates/gpec/) :

{% extends 'base.html' %}

{% block content %}

<h1>Contrôle Budgétaire des Salaires - {{ budget.annee }}</h1>

<div class="card mb-4">

<div class="card-body">

<h5 class="card-title">Vue d'ensemble</h5>

<p><strong>Budget total :</strong> {{ budget.budget\_total|floatformat:2 }} FCFA</p>

<p><strong>Masse salariale réelle :</strong> {{ masse\_salariale\_reelle|floatformat:2 }} FCFA</p>

<p><strong>Écart :</strong> {{ ecart|floatformat:2 }} FCFA</p>

</div>

</div>

<h2>Détail par département</h2>

<table class="table">

<thead>

<tr>

<th>Département</th>

<th>Budget alloué</th>

<th>Montant utilisé</th>

<th>Restant</th>

</tr>

</thead>

<tbody>

{% for budget\_dep in budgets\_departements %}

<tr>

<td>{{ budget\_dep.departement.denomination }}</td>

<td>{{ budget\_dep.montant\_alloue|floatformat:2 }} FCFA</td>

<td>{{ budget\_dep.montant\_utilise|floatformat:2 }} FCFA</td>

<td>{{ budget\_dep.montant\_restant|floatformat:2 }} FCFA</td>

</tr>

{% endfor %}

</tbody>

</table>

{% endblock %}

Étape 4 : Mise à jour des URLs

1. Ouvrez votre fichier urls.py et ajoutez la nouvelle URL :

from django.urls import path

from . import views

urlpatterns = [

# ... vos autres URLs ...

path('controle-budgetaire/', views.controle\_budgetaire, name='controle\_budgetaire'),

]

Étape 5 : Mise à jour du modèle Paiement

1. Modifiez votre modèle Paiement dans models.py pour mettre à jour automatiquement les budgets :

class Paiement(models.Model):

# ... vos champs existants ...

def save(self, \*args, \*\*kwargs):

super().save(\*args, \*\*kwargs)

# Mise à jour du budget

annee = self.date\_paiement.year

budget, created = BudgetSalarial.objects.get\_or\_create(annee=annee, defaults={'budget\_total': 0})

budget.budget\_utilise += self.salaire\_brut

budget.save()

# Mise à jour du budget départemental

departement = self.contrat.structure # Assurez-vous que ce chemin est correct

budget\_dep, created = BudgetDepartement.objects.get\_or\_create(

budget=budget,

departement=departement,

defaults={'montant\_alloue': 0}

)

budget\_dep.montant\_utilise += self.salaire\_brut

budget\_dep.save()

Étape 6 : Ajout d'un lien dans le menu de navigation

1. Modifiez votre template de base pour inclure un lien vers la nouvelle page de contrôle budgétaire :

<li class="nav-item">

<a class="nav-link {% if request.resolver\_match.url\_name == 'controle\_budgetaire' %}active{% endif %}" href="{% url 'controle\_budgetaire' %}">

<i class="fas fa-chart-pie"></i> Contrôle Budgétaire

</a>

</li>

Étape 7 : Testez et ajustez

1. Lancez votre serveur de développement et testez la nouvelle fonctionnalité.
2. Assurez-vous que les calculs sont corrects et que les mises à jour se font correctement.
3. Ajustez le design et les fonctionnalités selon vos besoins spécifiques.

Ces étapes vous permettront d'intégrer la fonctionnalité de contrôle budgétaire à votre système existant. N'oubliez pas d'adapter les chemins d'importation, les noms de modèles et les relations selon la structure exacte de votre projet.