**Modèle de Base de Données :** Créez un modèle dans Django pour représenter les écoles. Dans ce modèle, vous pouvez inclure un champ de notation qui stockera la note moyenne donnée par les utilisateurs.

from django.db import models

class Ecole(models.Model):

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

# Ajoutez d'autres champs nécessaires

note\_moyenne = models.FloatField(default=0.0)

**Modèle pour les Notations :** Créez un modèle pour stocker les notations données par les utilisateurs. Vous pouvez lier ce modèle à l'école notée et à l'utilisateur qui a donné la note.

from django.contrib.auth.models import User

class Notation(models.Model):

utilisateur = models.ForeignKey(User, on\_delete=models.CASCADE)

ecole = models.ForeignKey(Ecole, on\_delete=models.CASCADE)

note = models.IntegerField(choices=[(0, 'Nul'), (1, 'Médiocre'), (2, 'Passable'), (3, 'Assez bien'), (4, 'Bien'), (5, 'Excellente')])

**Vue pour la Notation :** Créez une vue pour gérer la notation des écoles. Cette vue permettra aux utilisateurs de sélectionner une école et de donner une note.

from django.shortcuts import render, redirect

from .models import Ecole, Notation

from django.contrib.auth.decorators import login\_required

@login\_required # Assurez-vous que l'utilisateur est connecté pour accéder à cette vue

def noter\_ecole(request, ecole\_id):

ecole = Ecole.objects.get(pk=ecole\_id)

if request.method == 'POST':

note = int(request.POST.get('note'))

notation = Notation.objects.create(utilisateur=request.user, ecole=ecole, note=note)

# Mettez à jour la note moyenne de l'école en fonction des nouvelles notations

ecole.update\_note\_moyenne()

return redirect('detail\_ecole', ecole\_id=ecole\_id)

**Mise à jour de la Note Moyenne :** Dans le modèle Ecole, vous pouvez ajouter une méthode pour mettre à jour la note moyenne de l'école en fonction des nouvelles notations.

class Ecole(models.Model):

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

note\_moyenne = models.FloatField(default=0.0)

def update\_note\_moyenne(self):

notations = self.notation\_set.all()

total\_notes = sum(n.note for n in notations)

nombre\_notations = notations.count()

if nombre\_notations > 0:

self.note\_moyenne = total\_notes / nombre\_notations

self.save()

**Affichage des Écoles avec leurs Notes :** Dans votre vue qui affiche les détails d'une école, vous pouvez inclure la note moyenne de l'école.

def detail\_ecole(request, ecole\_id):

ecole = Ecole.objects.get(pk=ecole\_id)

notations = Notation.objects.filter(ecole=ecole)

context = {

'ecole': ecole,

'notations': notations,

}

return render(request, 'detail\_ecole.html', context)

Dans le template **detail\_ecole.html**, vous pouvez afficher la note moyenne ainsi que les notations individuelles.