**Practical-6**

Aim: Apply concepts of Custom SQL query execution for all forms of the Social Media Application. Ensure that all queries (Select, Insert, Delete, Update etc.) are demonstrated.

**Code:**

Jobs/views.py:

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

from .models import JobPosting

from .forms import JobPostingForm

from django.db import connection

def create\_job\_posting(request):

    if request.method == 'POST':

        form = JobPostingForm(request.POST)

        if form.is\_valid():

            job\_posting = form.save(commit=False)

            job\_posting.save()

            return redirect('job\_postings')

    else:

        form = JobPostingForm()

    return render(request, 'create\_job.html', {'form': form})

def job\_postings(request):

    job\_postings = JobPosting.objects.all()

    return render(request,'jobPosting.html',{'job\_postings': job\_postings})

def job\_posting\_detail(request, job\_posting\_id):

    job\_posting = get\_object\_or\_404(JobPosting, pk=job\_posting\_id)

    return render(request, 'job\_posting\_detail.html', {'job\_posting': job\_posting})

def custom\_job\_posting(request):

    custom\_sql\_query = """

    SELECT id, title, description, created\_at FROM jobs\_jobposting ORDER BY created\_at DESC;"""

    with connection.cursor() as cursor:

        cursor.execute(custom\_sql\_query)

        columns = [col[0] for col in cursor.description]

        job\_postings = [dict(zip(columns, row)) for row in cursor.fetchall()]

    return render(request, 'custom\_job\_postings.html', {'job\_postings': job\_postings})

def add\_salary(request):

    with connection.cursor() as cursor:

        cursor.execute("UPDATE jobs\_jobposting SET salary = salary \* 1.10")

    return redirect('job\_postings')

def dec\_salary(request):

    with connection.cursor() as cursor:

        cursor.execute("UPDATE jobs\_jobposting SET salary = salary - (salary \* 0.10)")

    return redirect('job\_postings')

def delete\_job\_posting(request, job\_posting\_id):

    job\_posting = get\_object\_or\_404(JobPosting, pk=job\_posting\_id)

    if request.method == 'POST':

        with connection.cursor() as cursor:

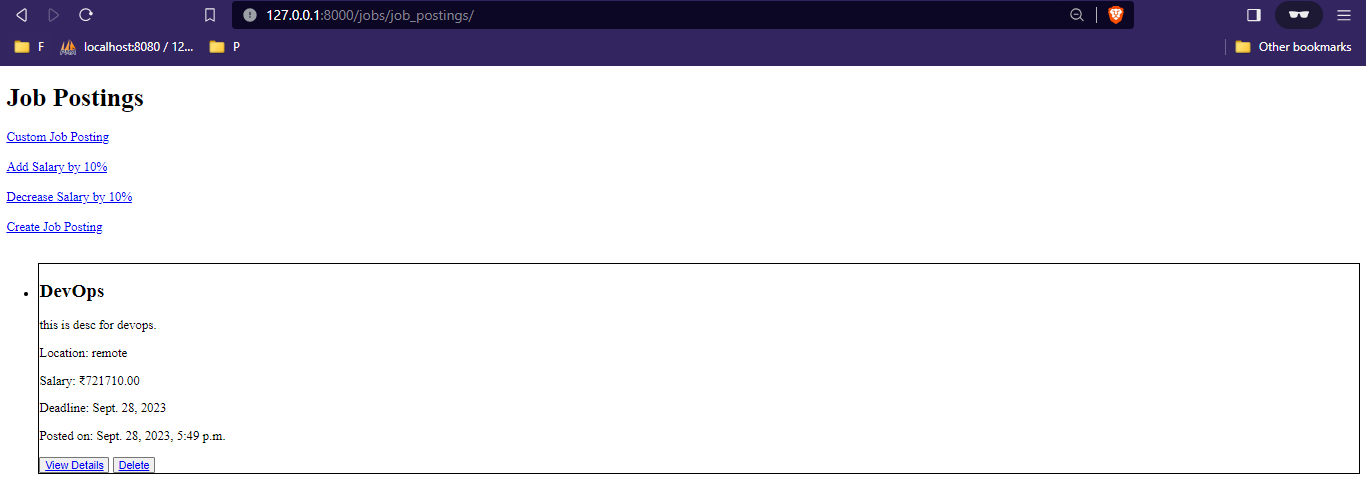
            cursor.execute("DELETE FROM jobs\_jobposting WHERE id = %s", [job\_posting\_id])

        return redirect('job\_postings')

    return render(request, 'delete\_job\_posting.html', {'job\_posting': job\_posting})

**Output:**

Job listing



After clicking on “Add Salary by 10%”.



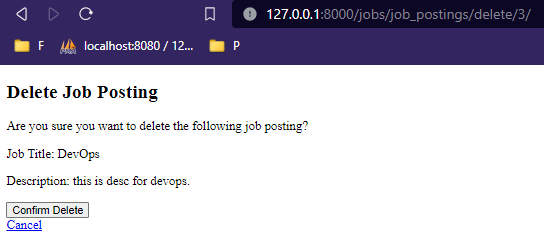
After clicking on “Decrease Salary by 10%”.



After clicking on “Custome Job Posting”.



After clicking on “Delete” button of DevOps job.



Confirm Delete.

