

## Practical-5

Aim: Creation of custom forms for the social media application. Apply concepts of Form Class, Model Form. Ensure appropriate models are considered for the various forms.

### Code:

jobs/models.py:

```
from django.db import models

class JobPosting(models.Model):
    title = models.CharField(max_length=100)
    description = models.TextField()
    location = models.CharField(max_length=100)
    salary = models.DecimalField(max_digits=10, decimal_places=2)
    deadline = models.DateField()
    created_at = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.title
```

jobs/forms.py:

```
from django import forms
from .models import JobPosting

class JobPostingForm(forms.ModelForm):
    class Meta:
        model = JobPosting
        fields = ['title', 'description', 'location', 'salary', 'deadline']

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)

        self.fields['description'].widget = forms.Textarea(attrs={'rows': 4})
```

jobs/urls.py:

```
from django.urls import path
from . import views

urlpatterns = [
    path('job_postings/', views.job_postings, name='job_postings'),
    path('create_job_posting/', views.create_job_posting, name='create_job_posting'),
    path('job_posting/<int:job_posting_id>/', views.job_posting_detail,
name='job_posting_detail'),
]
```

Jobs/views.py:

```
from django.shortcuts import render, redirect, get_object_or_404
from .models import JobPosting
from .forms import JobPostingForm

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()

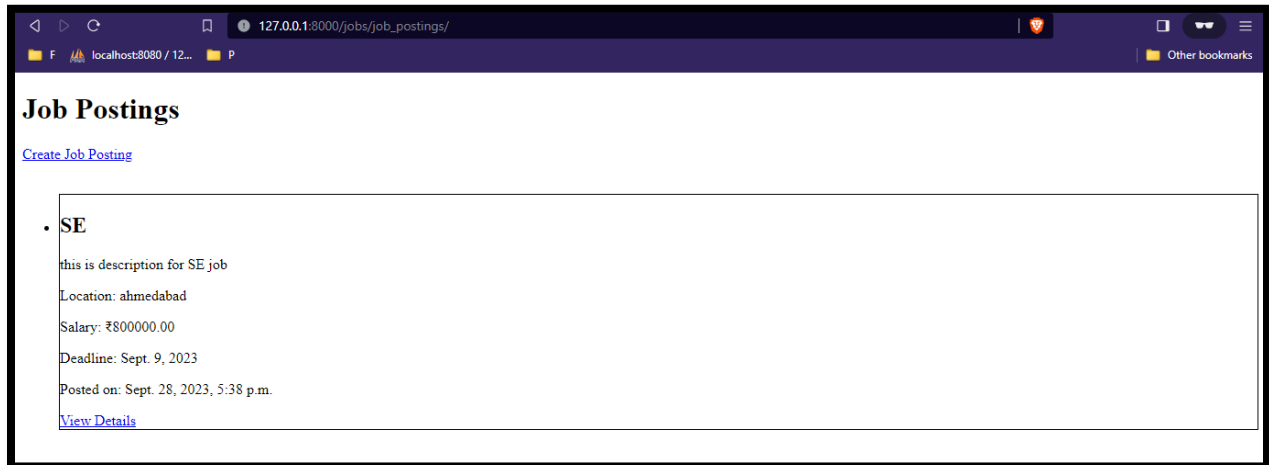
    return render(request, 'create_job.html', {'form': form})

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})
```

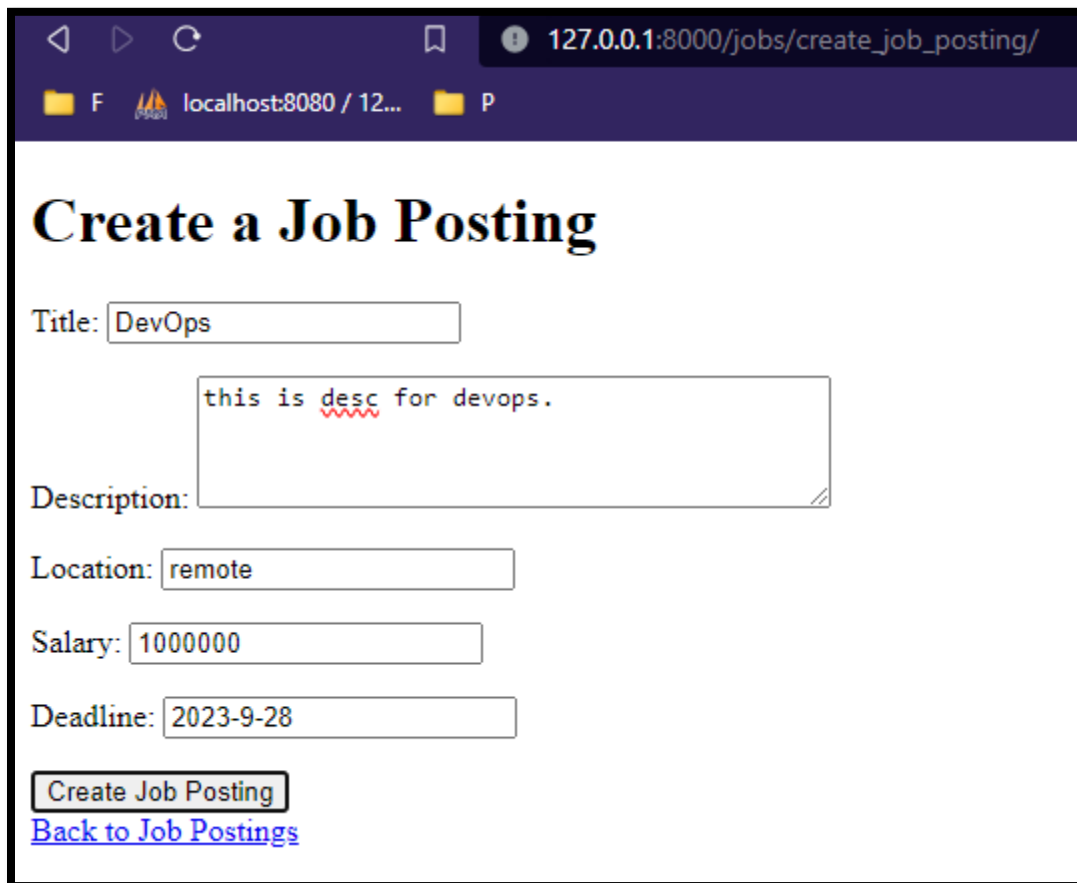
**Output:**

Home page (jobs posting):



After clicking on “Create Job Posting”.

Filling details.



127.0.0.1:8000/jobs/create\_job\_posting/

# Create a Job Posting

Title:

Description:

Location:

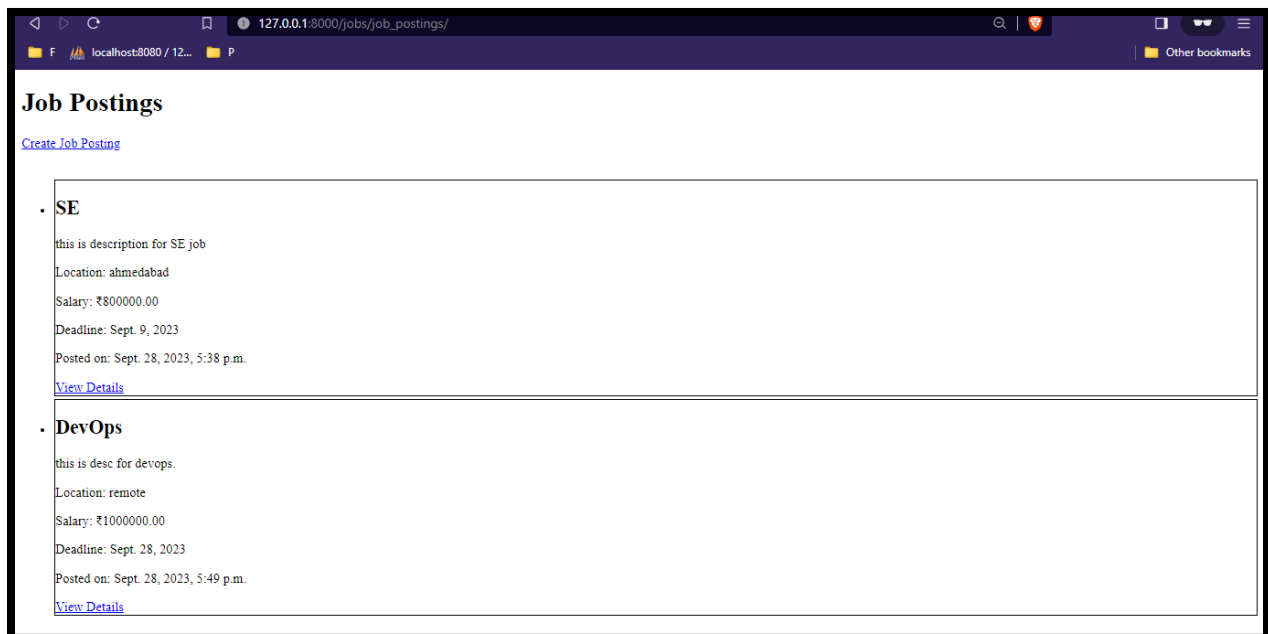
Salary:

Deadline:

[Create Job Posting](#)

[Back to Job Postings](#)

After clicking “Create Job Posting”.



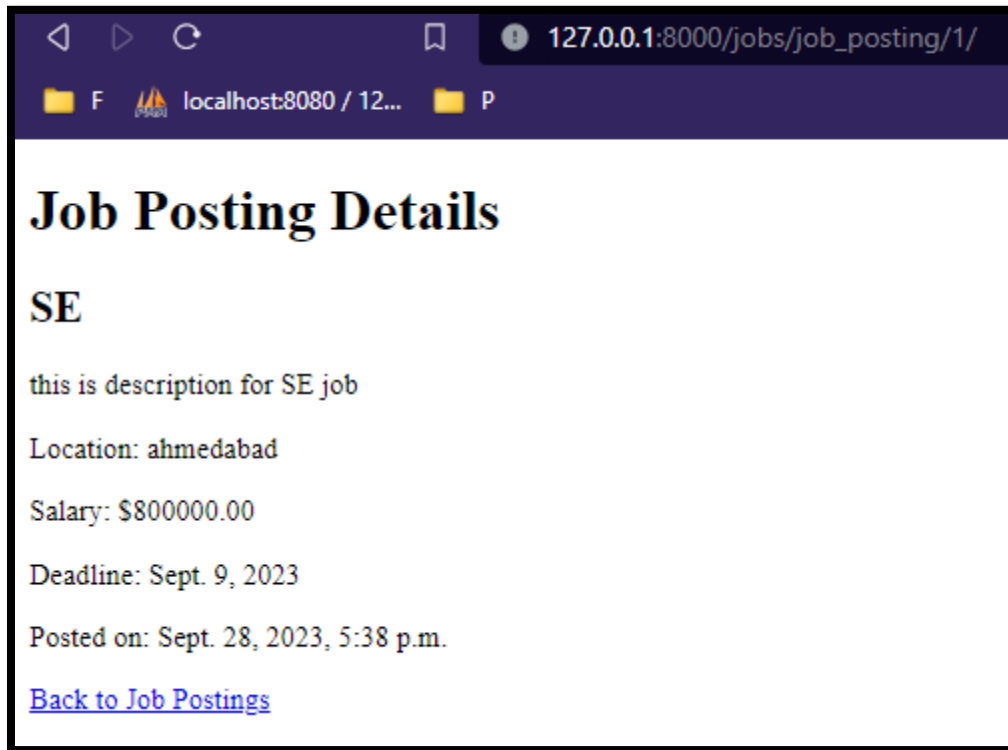
127.0.0.1:8000/jobs/job\_postings/

## Job Postings

[Create Job Posting](#)

- SE**  
this is description for SE job  
Location: ahmedabad  
Salary: ₹800000.00  
Deadline: Sept. 9, 2023  
Posted on: Sept. 28, 2023, 5:38 p.m.  
[View Details](#)
- DevOps**  
this is desc for devops.  
Location: remote  
Salary: ₹1000000.00  
Deadline: Sept. 28, 2023  
Posted on: Sept. 28, 2023, 5:49 p.m.  
[View Details](#)

Clicking on “View Details”.



Use of 404 not found:

In settings.py file

```
DEBUG = False

ALLOWED_HOSTS = ["http://127.0.0.1:8000/"]
```

In jobs/views.py file.

```
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})
```

