MYSORE UNIVERSITY SCHOOL OF ENGINEERING

Manasagangotri campus, Mysuru-570006
(Approved by AICTE, New Delhi)

UNIVERSITY OF MYSORE

An Assignment (21CD71, Full Stack Development) Report

On

"Implement a Secure Job Application Form using Django"

Submitted By

Monica Rose G

21SECD23

7th Semester,
Department of CS&D
MUSE.

Under Faculty Incharge

Dr. M. S. Govinde Gowda, Director.

Mr. Karthik M.N

Asst. Professor

Dept. of CS&D

MUSE

Job Application

Q4. Implement a Secure Job Application Form using Django where:

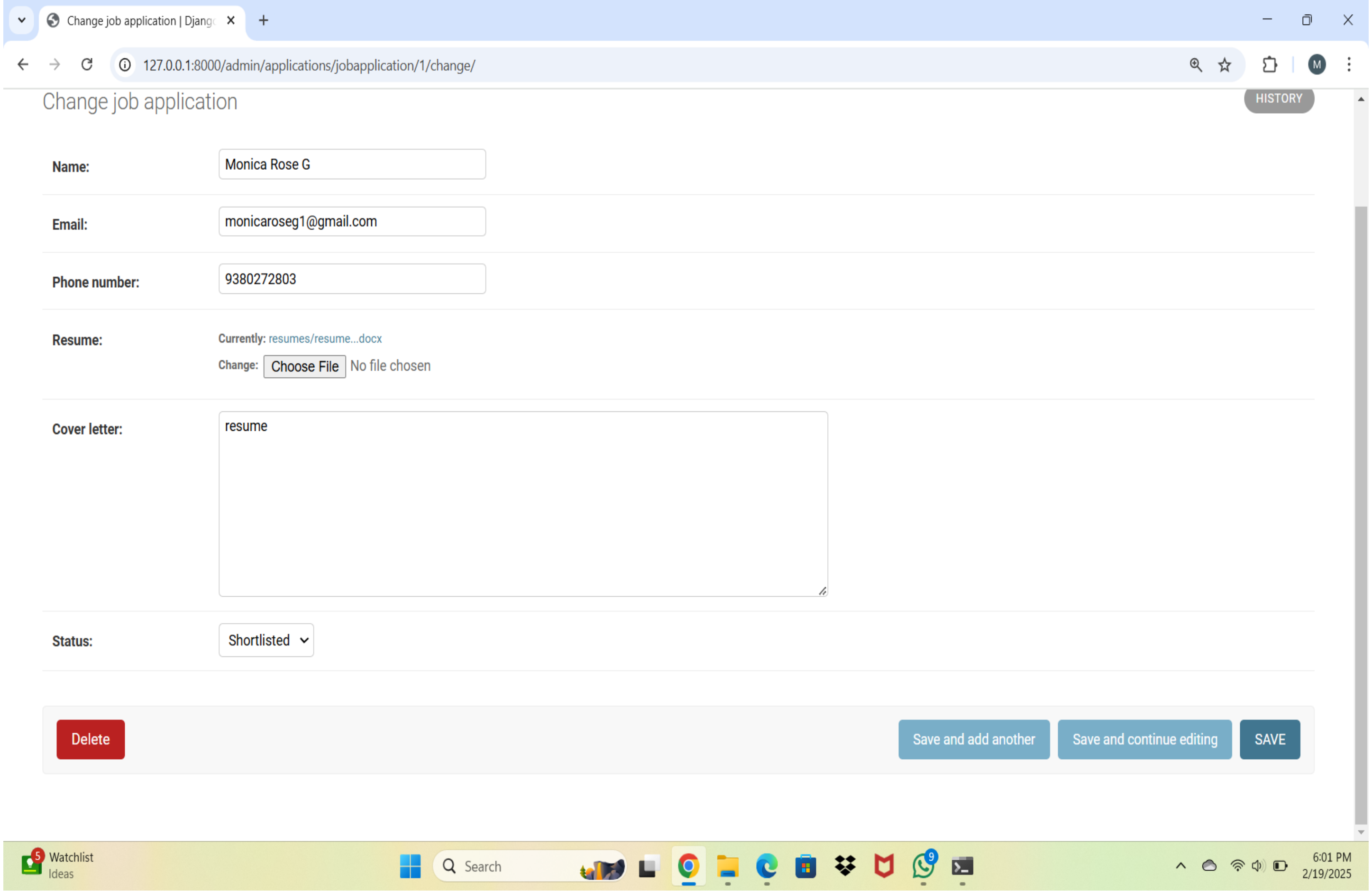
- Applicants can submit their name, email, phone number, resume (PDF only), and a cover letter.
- The application form should be implemented using Django ModelForms and should validate:
- The resume file is in PDF format only.
- The phone number contains exactly 10 digits.
- Store all applications in a database and allow the admin to view, shortlist, or reject applications.
- Implement CSRF protection for the form submission**.**

1.Introduction:

The Secure Job Application Form is a web-based system developed using Django to facilitate a streamlined and secure job application process. The system allows applicants to submit their details, ensuring data validation and security measures. Admins can review applications, shortlist, or reject candidates.

2. System Features

- Applicant Submission:
  - Name
  - Email
  - Phone number (10-digit validation)
  - Resume (PDF format only)
  - Cover Letter


Change job application | Djang
127.0.0.1:8000/admin/applications/jobapplication/1/change/
HISTORY
Choose File
Delete
Save and add another
Save and continue editing
SAVE
Watchlist
Ideas
Search
6:01 PM
2/19/2025

Change job application

Name: Monica Rose G

Email: monicaroseg1@gmail.com

Phone number: 9380272803

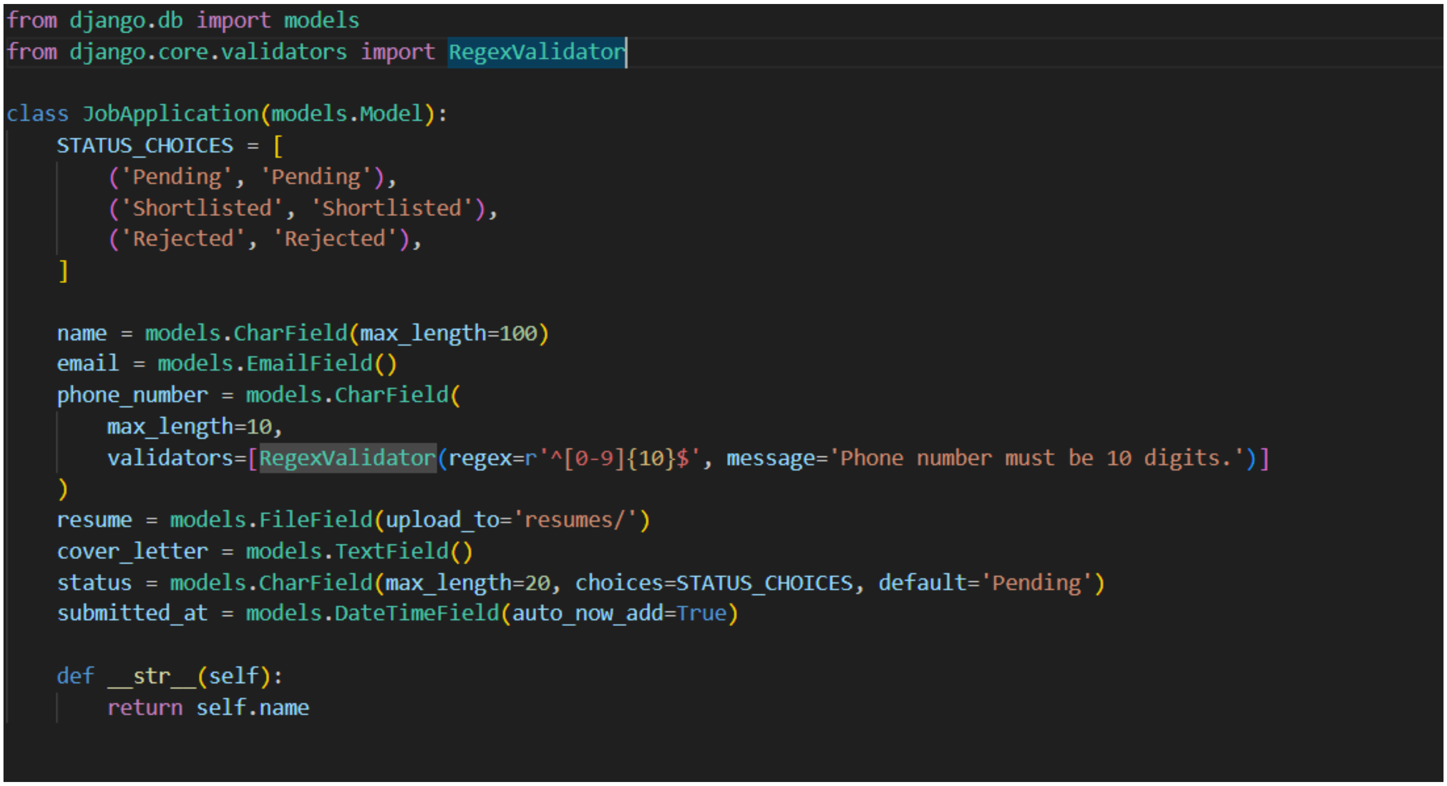
Resume: Currently: resumes/resume...docx
Change: No file chosen

Cover letter: resume

Status: Shortlisted

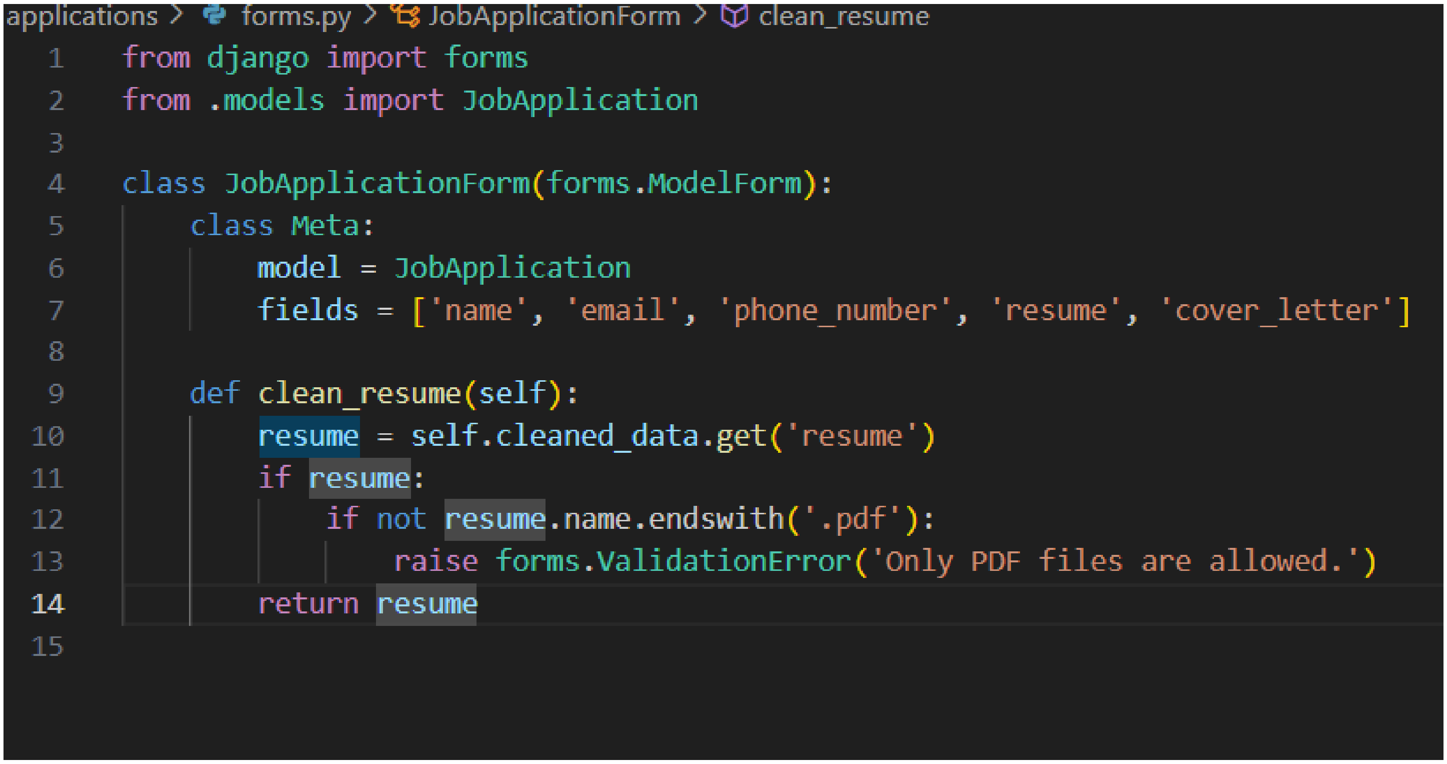
3. Implementation Details:

3.1: Model Design



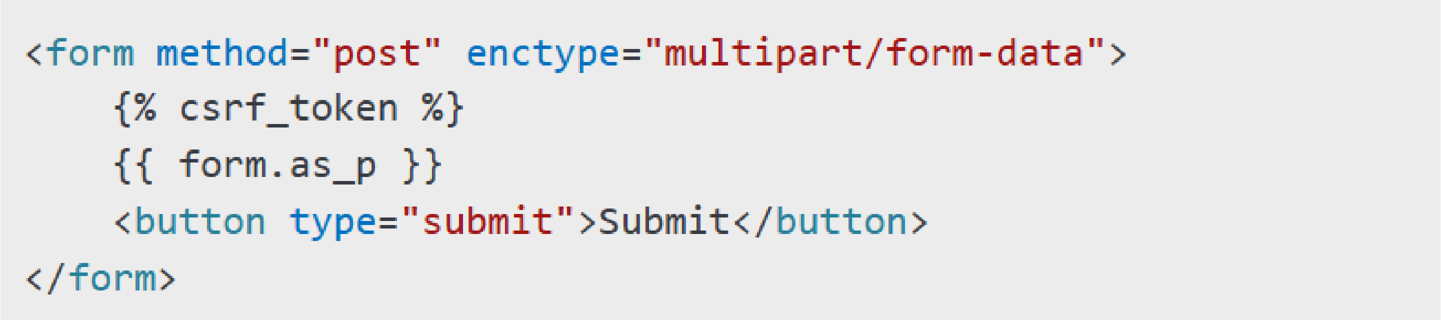
```python
from django.db import models
from django.core.validators import RegexValidator

class JobApplication(models.Model):
    STATUS_CHOICES = [
        ('Pending', 'Pending'),
        ('Shortlisted', 'Shortlisted'),
        ('Rejected', 'Rejected'),
    ]

    name = models.CharField(max_length=100)
    email = models.EmailField()
    phone_number = models.CharField(
        max_length=10,
        validators=[RegexValidator(regex=r'^[0-9]{10}$', message='Phone number must be 10 digits.')]
    )
    resume = models.FileField(upload_to='resumes/')
    cover_letter = models.TextField()
    status = models.CharField(max_length=20, choices=STATUS_CHOICES, default='Pending')
    submitted_at = models.DateTimeField(auto_now_add=True)

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

3.2: Form Validation



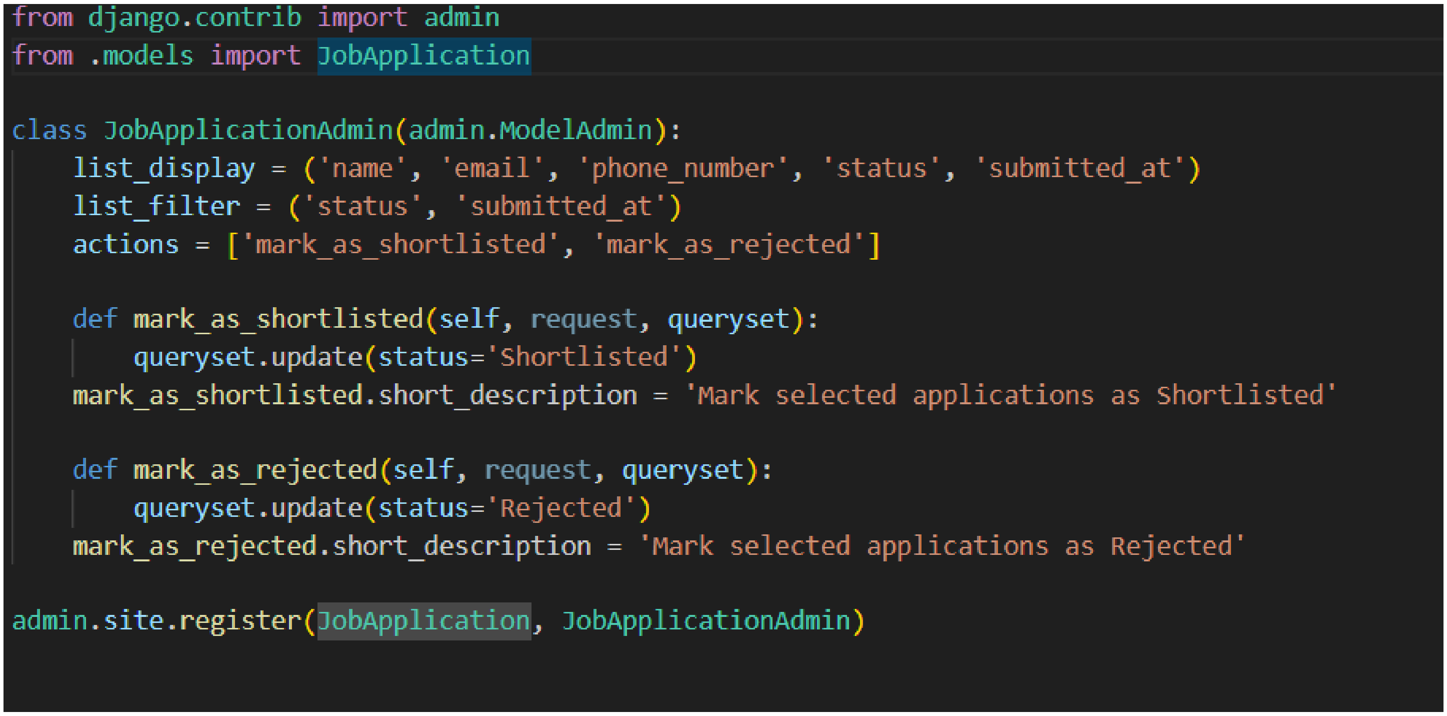
```python
from django import forms
from .models import JobApplication

class JobApplicationForm(forms.ModelForm):
    class Meta:
        model = JobApplication
        fields = ['name', 'email', 'phone_number', 'resume', 'cover_letter']

    def clean_resume(self):
        resume = self.cleaned_data.get('resume')
        if resume:
            if not resume.name.endswith('.pdf'):
                raise forms.ValidationError('Only PDF files are allowed.')
        return resume

```

3.3: CSRF Protection

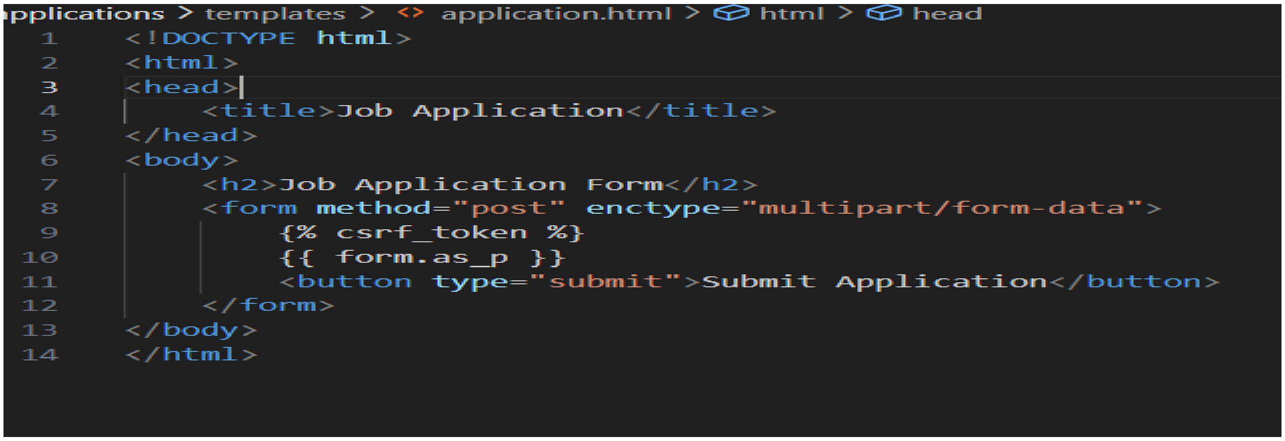


```
<form method="post" enctype="multipart/form-data">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Submit</button>
</form>
```

4. Admin Panel and Application Review

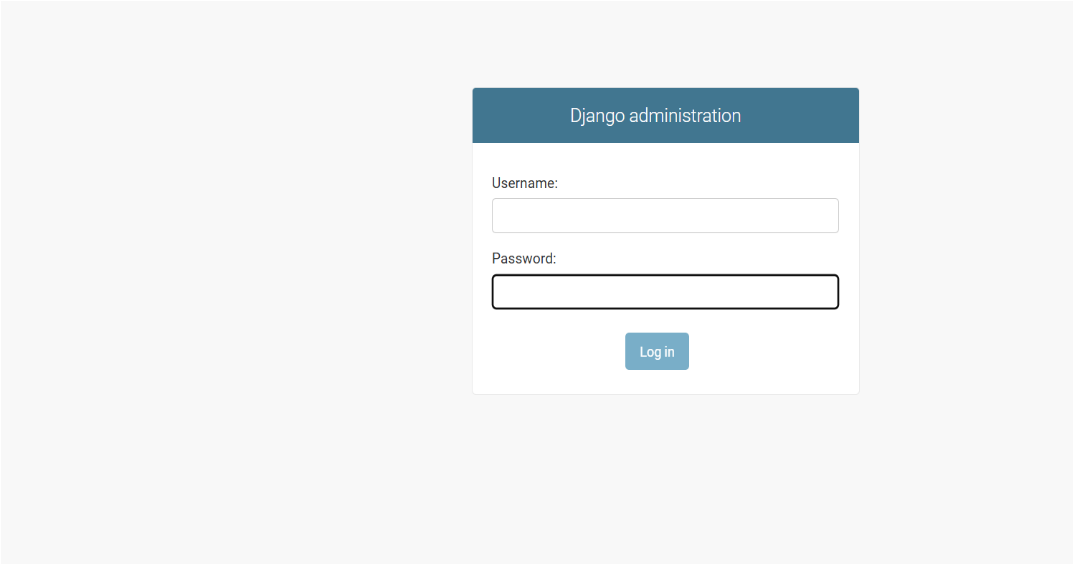


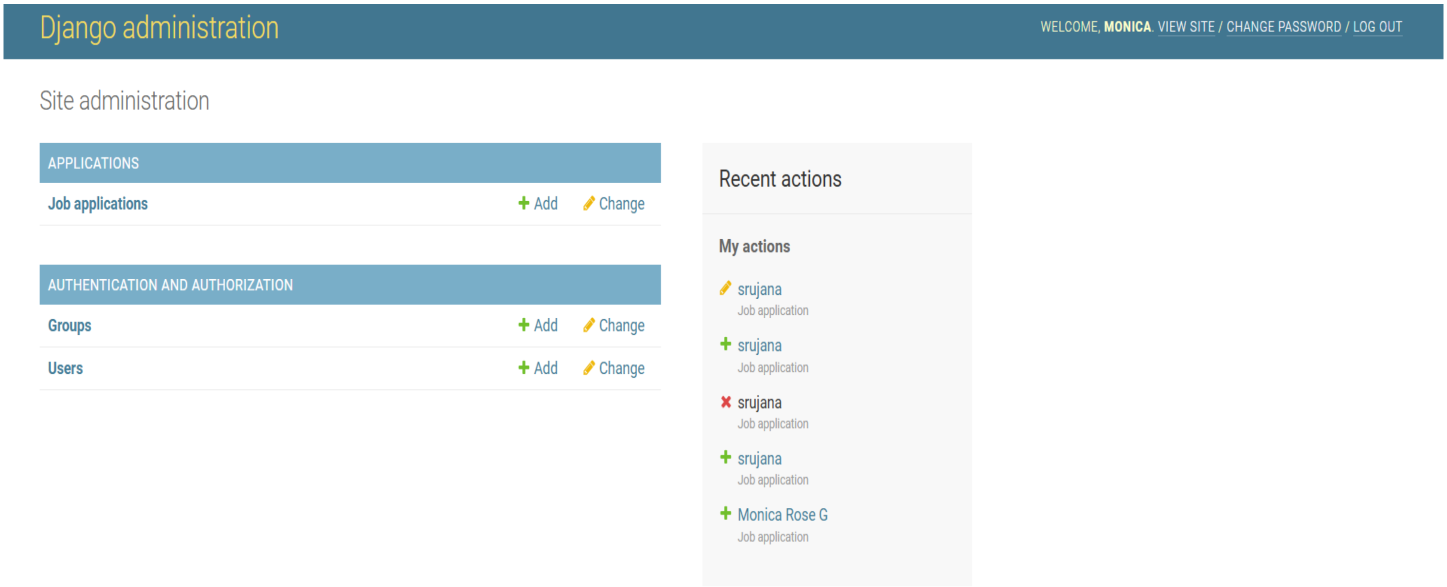
```python
from django.contrib import admin
from .models import JobApplication

class JobApplicationAdmin(admin.ModelAdmin):
    list_display = ('name', 'email', 'phone_number', 'status', 'submitted_at')
    list_filter = ('status', 'submitted_at')
    actions = ['mark_as_shortlisted', 'mark_as_rejected']

    def mark_as_shortlisted(self, request, queryset):
        queryset.update(status='Shortlisted')
    mark_as_shortlisted.short_description = 'Mark selected applications as Shortlisted'

    def mark_as_rejected(self, request, queryset):
        queryset.update(status='Rejected')
    mark_as_rejected.short_description = 'Mark selected applications as Rejected'

admin.site.register(JobApplication, JobApplicationAdmin)
```



```html
<!DOCTYPE html>
<html>
<head>
    <title>Job Application</title>
</head>
<body>
    <h2>Job Application Form</h2>
    <form method="post" enctype="multipart/form-data">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Submit Application</button>
    </form>
</body>
</html>
```

Output Link: http://127.0.0.1:8000/admin/login/?next=/admin/





Site administration

APPLICATIONS

Job applications ✚ Add ✎ Change

AUTHENTICATION AND AUTHORIZATION

Groups ✚ Add ✎ Change

Users ✚ Add ✎ Change

Recent actions

My actions

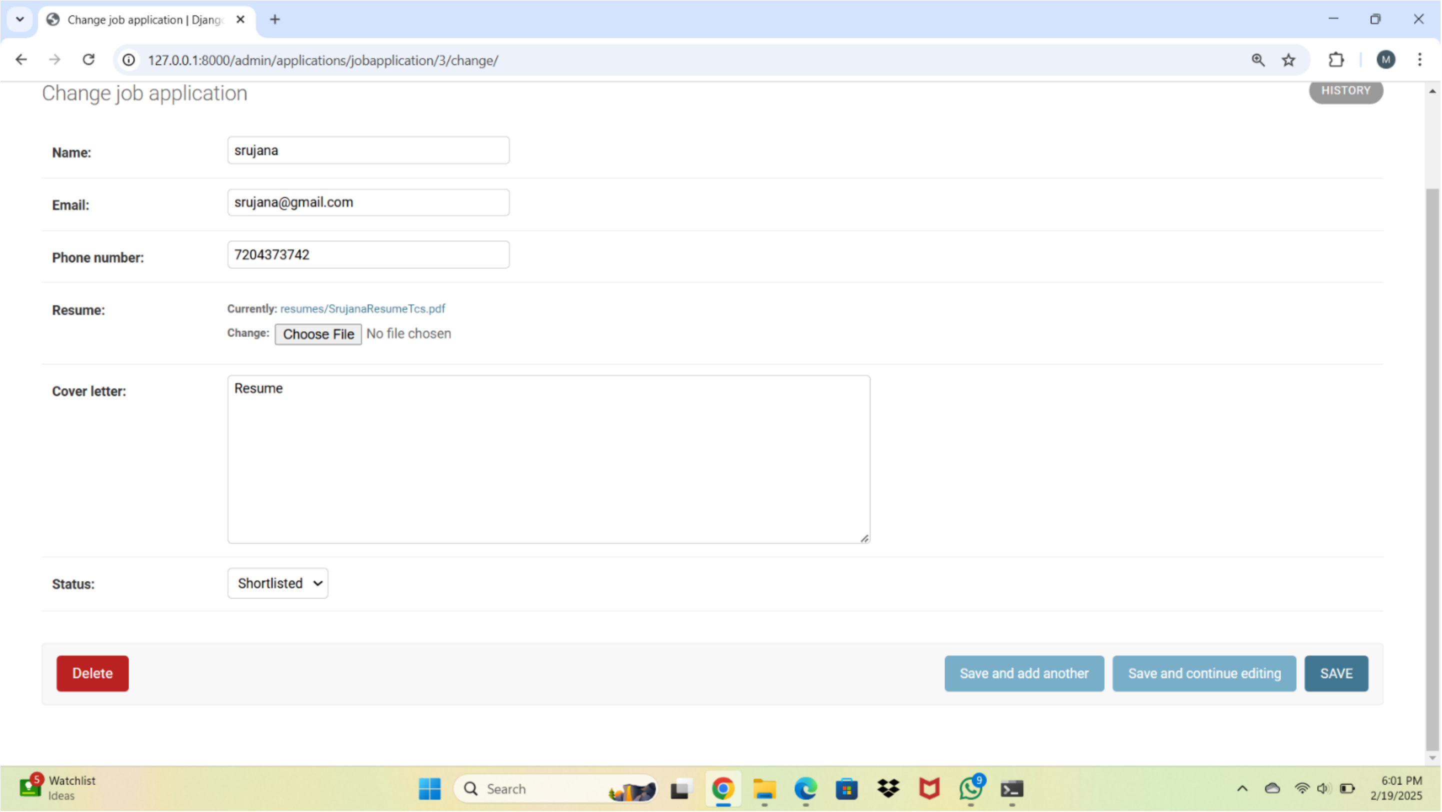
✎ srujana
Job application

✚ srujana
Job application

✖ srujana
Job application

✚ srujana
Job application

✚ Monica Rose G
Job application

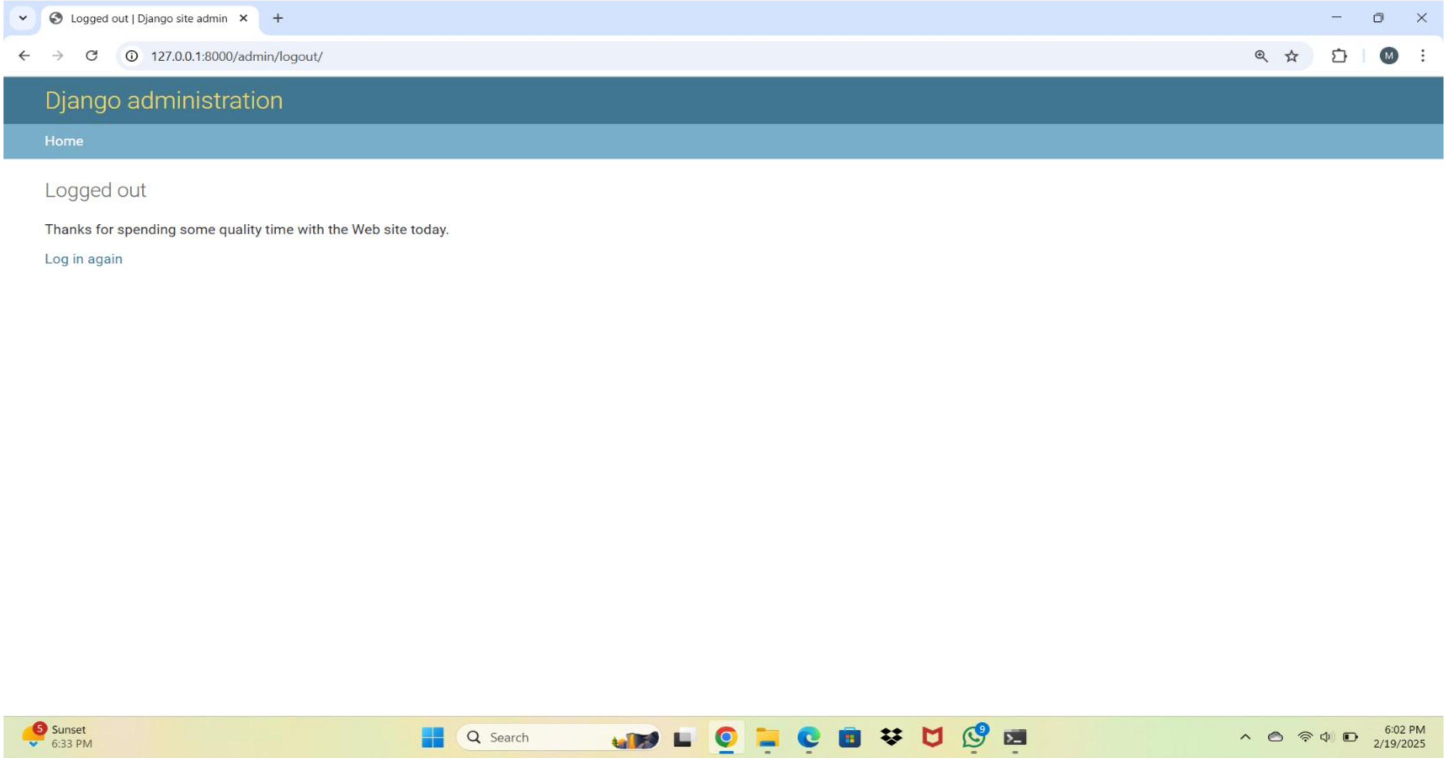


Change job application | Djang
127.0.0.1:8000/admin/applications/jobapplication/3/change/
HISTORY

Change job application

Name:	srujana
Email:	srujana@gmail.com
Phone number:	7204373742
Resume:	Currently: resumes/SrujanaResumeTcs.pdf Change: Choose File No file chosen
Cover letter:	Resume
Status:	Shortlisted

Delete
Save and add another
Save and continue editing
SAVE
Watchlist
Ideas
Search
6:01 PM
2/19/2025



Logged out

Thanks for spending some quality time with the Web site today.

Log in again

5. Conclusion

The Secure Job Application Form ensures data integrity, security, and ease of applicant management. Features like CSRF protection, file validation, and form validation enhance the security of the system, making it a robust solution for job application management.

6. Github Repo Link

https://github.com/RoseMonicaG/Job-application-using-Django