**Mastering Django Form read complete at the end**

Video lecture 45

Code file 38

MAKE MEDIA AND ADD THINGS IN THAT :

Settings.py:

STATIC\_URL = 'static/'

MEDIA\_URL = '/media/'  # for downloading images or doc from database

MEDIA\_ROOT = BASE\_DIR / 'media'

AND ADD THESE THINGS IN URLS.PY: do these things for downloading :

from django.conf import settings

from django.conf.urls.static import static

urlpatterns = [

    path('admin/', admin.site.urls),

    path('', include('student.urls')),

] + static(settings.MEDIA\_URL,document\_root = settings.MEDIA\_ROOT)

First we shows llike :



Than we can remove the list like []or other things we use templatetags (custom filter)

Customfilter.py:

from django import templates

register = templates.Library()

@register.filter(name='remove\_chars')

def remove\_chars(value,arg):

    for character in arg:

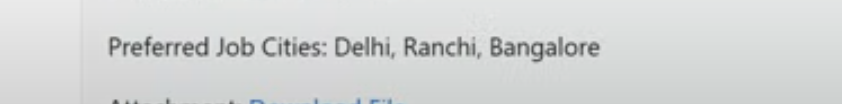
        value = value.replace(character,"")

    return value

and use in html:

<p>Preferred Job Cities: {{candidate.job\_city|remove\_char:"[]'"}}</p>

And now its look like :



NOW OTHER COMPLETE CODE FILES

Models.py:

from django.db import models

from django.core.exceptions import ValidationError   # Used for custom field validation

from django.core.validators import RegexValidator    # Used for validating mobile number format

# -----------------------------

# Custom Validator Functions

# -----------------------------

def validate\_pin\_address(value):

    """

    Custom validator for 'pin' field.

    Ensures that the PIN code must be exactly 6 digits long.

    """

    if len(str(value)) != 6:

        raise ValidationError('The PIN code must be exactly 6 digits.')

# -----------------------------

# Example Hardcoded Choices (Indian States)

# Later you can replace with Pakistani Provinces or load dynamically from API/DB.

# -----------------------------

STATE\_CHOICES = (

    ('Andhra Pradesh', 'Andhra Pradesh'),

    ('Arunachal Pradesh', 'Arunachal Pradesh'),

    ('Assam', 'Assam'),

    ('Bihar', 'Bihar'),

    ('Chhattisgarh', 'Chhattisgarh'),

    ('Goa', 'Goa'),

    ('Gujarat', 'Gujarat'),

    ('Haryana', 'Haryana'),

    ('Himachal Pradesh', 'Himachal Pradesh'),

    ('Jharkhand', 'Jharkhand'),

    ('Karnataka', 'Karnataka'),

    ('Kerala', 'Kerala'),

    ('Madhya Pradesh', 'Madhya Pradesh'),

    ('Maharashtra', 'Maharashtra'),

    ('Manipur', 'Manipur'),

    ('Meghalaya', 'Meghalaya'),

    ('Mizoram', 'Mizoram'),

    ('Nagaland', 'Nagaland'),

    ('Odisha', 'Odisha'),

    ('Punjab', 'Punjab'),

    ('Rajasthan', 'Rajasthan'),

    ('Sikkim', 'Sikkim'),

    ('Tamil Nadu', 'Tamil Nadu'),

    ('Telangana', 'Telangana'),

    ('Tripura', 'Tripura'),

    ('Uttar Pradesh', 'Uttar Pradesh'),

    ('Uttarakhand', 'Uttarakhand'),

    ('West Bengal', 'West Bengal'),

    ('Andaman and Nicobar Islands', 'Andaman and Nicobar Islands'),

    ('Chandigarh', 'Chandigarh'),

    ('Dadra and Nagar Haveli and Daman and Diu', 'Dadra and Nagar Haveli and Daman and Diu'),

    ('Delhi', 'Delhi'),

    ('Jammu and Kashmir', 'Jammu and Kashmir'),

    ('Ladakh', 'Ladakh'),

    ('Lakshadweep', 'Lakshadweep'),

    ('Puducherry', 'Puducherry'),

)

# -----------------------------

# Profile Model

# -----------------------------

class Profile(models.Model):

    """

    Profile model to store user details.

    Includes validation for PIN code, mobile number, and email.

    """

    # Basic Information

    name = models.CharField(max\_length=100)               # Full name of the person

    dob = models.DateField()                              # Date of birth

    gender = models.CharField(max\_length=1)               # Gender (M/F/Other)

    # Address Information

    locality = models.CharField(max\_length=100)           # Local area or street name

    city = models.CharField(max\_length=100)               # City name

    pin = models.PositiveIntegerField(                    # PIN code (exactly 6 digits)

        validators=[validate\_pin\_address],

        help\_text='Enter a 6-digit PIN code'

    )

    state = models.CharField(                             # State field (dropdown choices)

        choices=STATE\_CHOICES,

        max\_length=100

    )

    # Contact Information

    mobile = models.CharField(                            # Mobile number (must be 11 digits)

        max\_length=11,

        validators=[RegexValidator(regex=r'^\d{11}$', message="Mobile number must be 11 digits")],

        help\_text='Enter an 11-digit mobile number'

    )

    email = models.EmailField()                           # Email address (with proper validation)

    # Job / Other Info

    job\_city = models.CharField(max\_length=50)            # Preferred job location

    # Media Fields

    profile\_image = models.ImageField(                    # Profile photo (optional)

        upload\_to='profileimg',

        blank=True,

        null=True

    )

    my\_file = models.FileField(                           # File upload (optional)

        upload\_to='doc',

        blank=True,

        null=True

    )

    def \_\_str\_\_(self):

        """String representation of the model"""

        return self.name

Forms.py:

from django import forms

from student.models import Profile

GENDER\_CHOICES = (

    ('M', 'Male'),

    ('F', 'Female'),

    ('O', 'Other'),

)

JOB\_CITY\_CHOICE = [

    ('Delhi', 'Delhi'),

    ('Pune', 'Pune'),

    ('Ranchi', 'Ranchi'),

    ('Mumbai', 'Mumbai'),

    ('Dhanbad', 'Dhanbad'),

    ('Bangalore', 'Bangalore'),

]

class ProfileForm(forms.ModelForm):

    gender = forms.ChoiceField(widget=forms.RadioSelect, choices=GENDER\_CHOICES, required=False)

    job\_city = forms.MultipleChoiceField(choices = JOB\_CITY\_CHOICE) #here is 1 method to choose different multiple fileds

    #here are the second method

    job\_city = forms.MultipleChoiceField(choices = JOB\_CITY\_CHOICE,

                                         widget=forms.CheckboxSelectMultiple,

                                         label='prefered Job Cities',

                                         help\_text='Select 1 or more cities where you prefered to work')

    class Meta:

        model = Profile

        fields = [

            'name', 'dob', 'gender', 'locality', 'city', 'pin', 'state', 'mobile', 'email', 'job\_city', 'profile\_image', 'my\_file'

        ]

        labels = {

            'name': 'Full Name',

            'dob': 'Date of Birth',

            'pin': 'Pin Code',

            'mobile': 'Mobile Number'

        }

        help\_texts = {

            'profile\_image': 'Optional: Upload a profile image',

            'my\_file': 'Optional: Attach any additional document (PDF, DOCX, etc.)'

        }

        widgets = {

            'name' :forms.TextInput(attrs={'class':'form-control'}),

            'dob'  :forms.DateInput(attrs={'class':'form-control','id':'datepicker','type':'date'}),

            'locality': forms.TextInput(attrs={'class':'form-control','placeholder':'Write your area name '}),

            'city':forms.TextInput(attrs={'class':'form-control','placeholder':'city name'}),

            'pin':forms.NumberInput(attrs={'class':'form-control','placeholder':'Write your area name '}),

            'state': forms.Select(attrs={'class': 'form-control'}),

            'mobile':forms.TextInput(attrs={'class':'form-control','placeholder':'11-digit Mobile Number'}),

            'email' :forms.EmailInput(attrs={'class':'form-control','placeholder':'xyz@example.com'})

        }

Views.py:

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

from student.forms import ProfileForm

from student.models import Profile

# Home view - show all candidates and handle form submission

def Home(request):

    candidates = Profile.objects.all()

    if request.method == 'POST':

        form = ProfileForm(request.POST, request.FILES)

        if form.is\_valid():

            form.save()

            return redirect('Home')  # make sure 'Home' is the correct URL name

    else:

        form = ProfileForm()  # only create a blank form if GET request

    return render(request, 'student/home.html', {'form': form, 'candidates': candidates})

# Candidate detail view - show details of one candidate

def candidate\_detail(request, pk):

    candidate = get\_object\_or\_404(Profile, pk=pk)  # safer than Profile.objects.get()

    return render(request, 'student/candidate.html', {'candidate': candidate})

base.html:

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Document</title>

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">

</head>

<body>

  <div class="container">

    <div class="text-center mt-3">

      <h1>Mastering Django Form Fields</h1>

    </div>

    <div class="row my-5">

      <!-- Column 1 -->

      <div class="col-md-8">

         <div class='p-3 border bg-light'>

          {% block left-content %}{% endblock left-content %}

         </div>

      </div>

      <!--Column 2-->

      <div class="col-md-4">

        <div class='p-3 border bg-light'>

         {% block right-content %}{% endblock right-content %}

        </div>

     </div>

    </div>

  </div>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>

</body>

</html>

Candiate.html:

{% extends "student/base.html" %}

{% load customfilter %}

{% block left-content %}

<h1>Candidate Detail</h1>

<p class="float-end">Application No: {{candidate.id}}</p>

<img src="{{candidate.profile\_image.url}}" alt="" class="img-fluid img-thumbnail float-end mt-4" width="150" height="150">

<p>Name: {{candidate.name}}</p>

<p>Date of birth: {{candidate.dob}}</p>

<p>Gender: {{candidate.gender}}</p>

<p>Address: {{candidate.locality}}, {{candidate.city}}, {{candidate.state}} - {{candidate.pin}}</p>

<p>Mobile No: {{candidate.mobile}}</p>

<p>Email ID: {{candidate.email}}</p>

<p>Preferred Job Cities: {{candidate.job\_city|remove\_chars:"[]'"}}</p>

<p>Attachment: <a href="{{candidate.my\_file.url}}">Download File</a></p>

<div class="text-center">

  <a href="/" class="btn btn-danger">Back to Home</a>

</div>

{% endblock left-content %}

Home.html:

{% extends "student/base.html" %}

{% block left-content %}

<form action="" method="POST" enctype="multipart/form-data">

  {% csrf\_token %}

  <!-- Non field errors -->

  {% if form.non\_field\_errors %}

    {% for error in form.non\_field\_errors %}

      <p class="alert alert-danger my-3">{{ error }}</p>

    {% endfor %}

  {% endif %}

  <!-- Name -->

  <div class="mb-3">

    <label for="{{ form.name.id\_for\_label }}" class="form-label">{{ form.name.label }}</label>

    {{ form.name }}

    {% for error in form.name.errors %}

      <p class="text-danger">{{ error }}</p>

    {% endfor %}

  </div>

  <!-- DOB -->

  <div class="mb-3">

    <label for="{{ form.dob.id\_for\_label }}" class="form-label">{{ form.dob.label }}</label>

    {{ form.dob }}

    {% for error in form.dob.errors %}

      <p class="text-danger">{{ error }}</p>

    {% endfor %}

  </div>

  <!-- Gender -->

  <div class="mb-3">

    <label for="{{ form.gender.id\_for\_label }}" class="form-label">{{ form.gender.label }}</label>

    {{ form.gender }}

    {% for error in form.gender.errors %}

      <p class="text-danger">{{ error }}</p>

    {% endfor %}

  </div>

  <!-- Locality -->

  <div class="mb-3">

    <label for="{{ form.locality.id\_for\_label }}" class="form-label">{{ form.locality.label }}</label>

    {{ form.locality }}

    {% for error in form.locality.errors %}

      <p class="text-danger">{{ error }}</p>

    {% endfor %}

  </div>

  <!-- City -->

  <div class="mb-3">

    <label for="{{ form.city.id\_for\_label }}" class="form-label">{{ form.city.label }}</label>

    {{ form.city }}

    {% for error in form.city.errors %}

      <p class="text-danger">{{ error }}</p>

    {% endfor %}

  </div>

  <!-- Pin -->

  <div class="mb-3">

    <label for="{{ form.pin.id\_for\_label }}" class="form-label">{{ form.pin.label }}</label>

    {{ form.pin }}

    {% if form.pin.help\_text %}

      <div class="form-text">{{ form.pin.help\_text }}</div>

    {% endif %}

    {% for error in form.pin.errors %}

      <p class="text-danger">{{ error }}</p>

    {% endfor %}

  </div>

  <!-- State -->

  <div class="mb-3">

    <label for="{{ form.state.id\_for\_label }}" class="form-label">{{ form.state.label }}</label>

    {{ form.state }}

    {% for error in form.state.errors %}

      <p class="text-danger">{{ error }}</p>

    {% endfor %}

  </div>

  <!-- Mobile -->

  <div class="mb-3">

    <label for="{{ form.mobile.id\_for\_label }}" class="form-label">{{ form.mobile.label }}</label>

    {{ form.mobile }}

    {% if form.mobile.help\_text %}

      <div class="form-text">{{ form.mobile.help\_text }}</div>

    {% endif %}

    {% for error in form.mobile.errors %}

      <p class="text-danger">{{ error }}</p>

    {% endfor %}

  </div>

  <!-- Email -->

  <div class="mb-3">

    <label for="{{ form.email.id\_for\_label }}" class="form-label">{{ form.email.label }}</label>

    {{ form.email }}

    {% for error in form.email.errors %}

      <p class="text-danger">{{ error }}</p>

    {% endfor %}

  </div>

  <!-- Job City -->

  <div class="mb-3">

    <label for="{{ form.job\_city.id\_for\_label }}" class="form-label">{{ form.job\_city.label }}</label>

    {{ form.job\_city }}

    {% if form.job\_city.help\_text %}

      <div class="form-text">{{ form.job\_city.help\_text }}</div>

    {% endif %}

    {% for error in form.job\_city.errors %}

      <p class="text-danger">{{ error }}</p>

    {% endfor %}

  </div>

  <!-- Profile Image -->

  <div class="mb-3">

    <label for="{{ form.profile\_image.id\_for\_label }}" class="form-label">{{ form.profile\_image.label }}</label>

    {{ form.profile\_image }}

    {% if form.profile\_image.help\_text %}

      <div class="form-text">{{ form.profile\_image.help\_text }}</div>

    {% endif %}

    <div class="mt-3">

      <img id="imagePreview" src="" alt="Image Preview" style="display:none; max-width:50%; height:auto;">

    </div>

    {% for error in form.profile\_image.errors %}

      <p class="text-danger">{{ error }}</p>

    {% endfor %}

  </div>

  <!-- My File -->

  <div class="mb-3">

    <label for="{{ form.my\_file.id\_for\_label }}" class="form-label">{{ form.my\_file.label }}</label>

    {{ form.my\_file }}

    {% if form.my\_file.help\_text %}

      <div class="form-text">{{ form.my\_file.help\_text }}</div>

    {% endif %}

    {% for error in form.my\_file.errors %}

      <p class="text-danger">{{ error }}</p>

    {% endfor %}

  </div>

  <!-- Submit button -->

  <button type="submit" class="btn btn-primary">Submit</button>

</form>

<!-- Script for image preview -->

<script>

  function previewImage(event) {

    const imagePreview = document.getElementById('imagePreview');

    const file = event.target.files[0];

    if (file) {

      const reader = new FileReader();

      reader.onload = function(e) {

        imagePreview.src = e.target.result;

        imagePreview.style.display = 'block';

      };

      reader.readAsDataURL(file);

    } else {

      imagePreview.style.display = 'none';

    }

  }

  const profileInput = document.getElementById("{{ form.profile\_image.id\_for\_label }}");

  if (profileInput) {

    profileInput.addEventListener("change", previewImage);

  }

</script>

{% endblock left-content %}

{% block right-content %}

<h4 class="text-center mb-3"><u>List of Candidate</u></h4>

{% for candidate in candidates %}

  <p>{{ candidate.id }} : <a href="{% url 'candidate\_detail' candidate.id %}">{{ candidate.name }}</a></p>

{% empty %}

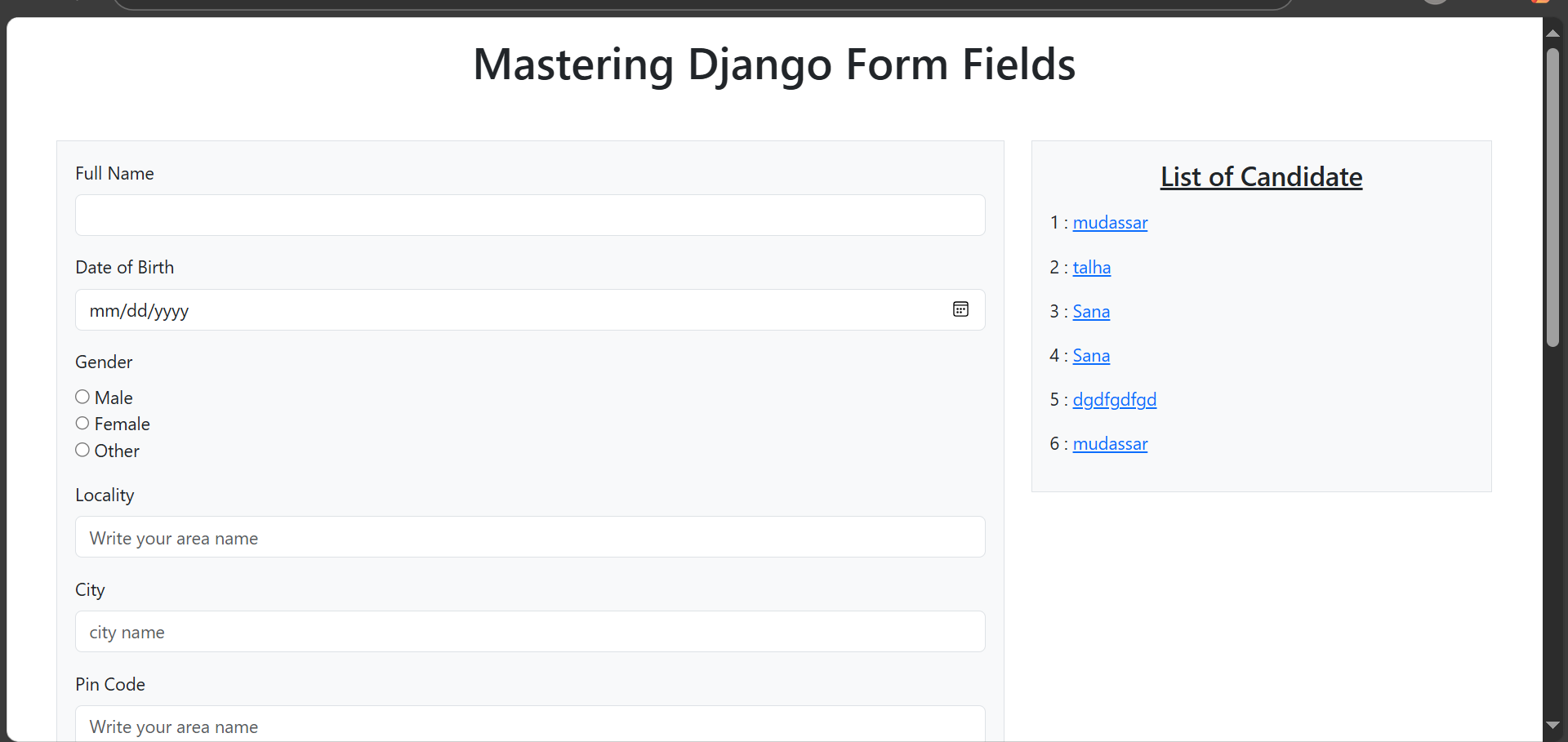
  <p class="text-muted">No candidates available.</p>

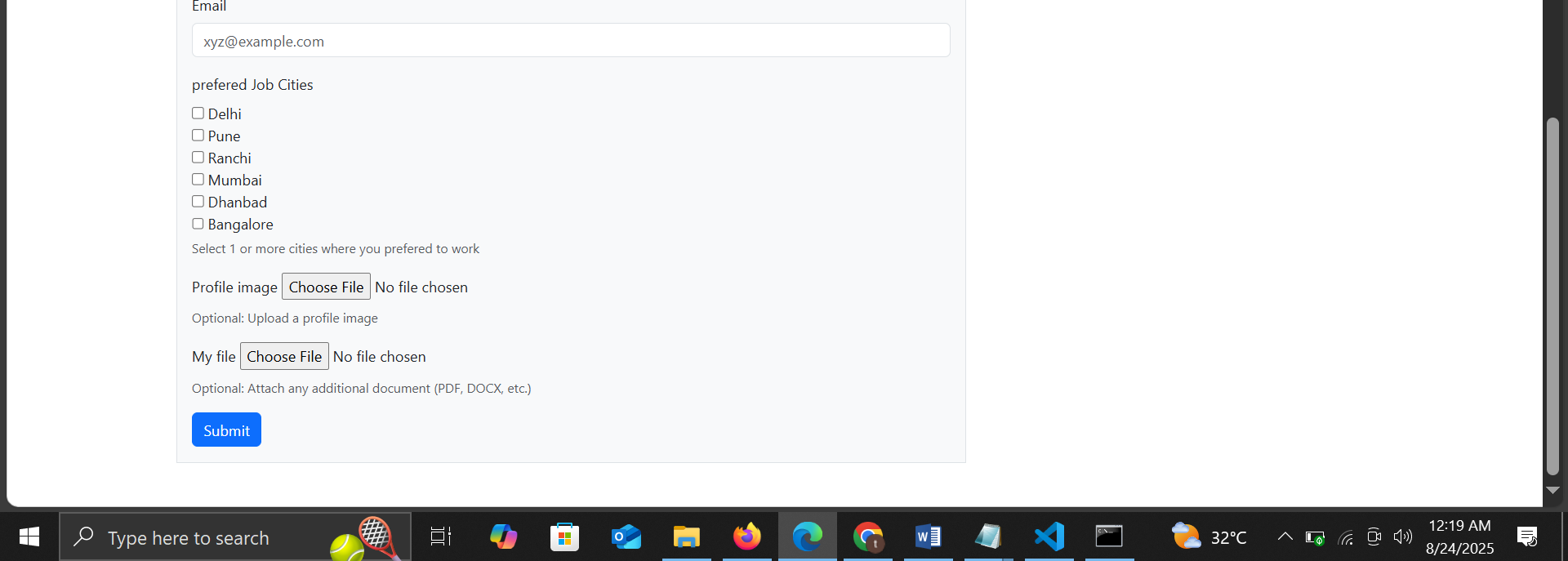
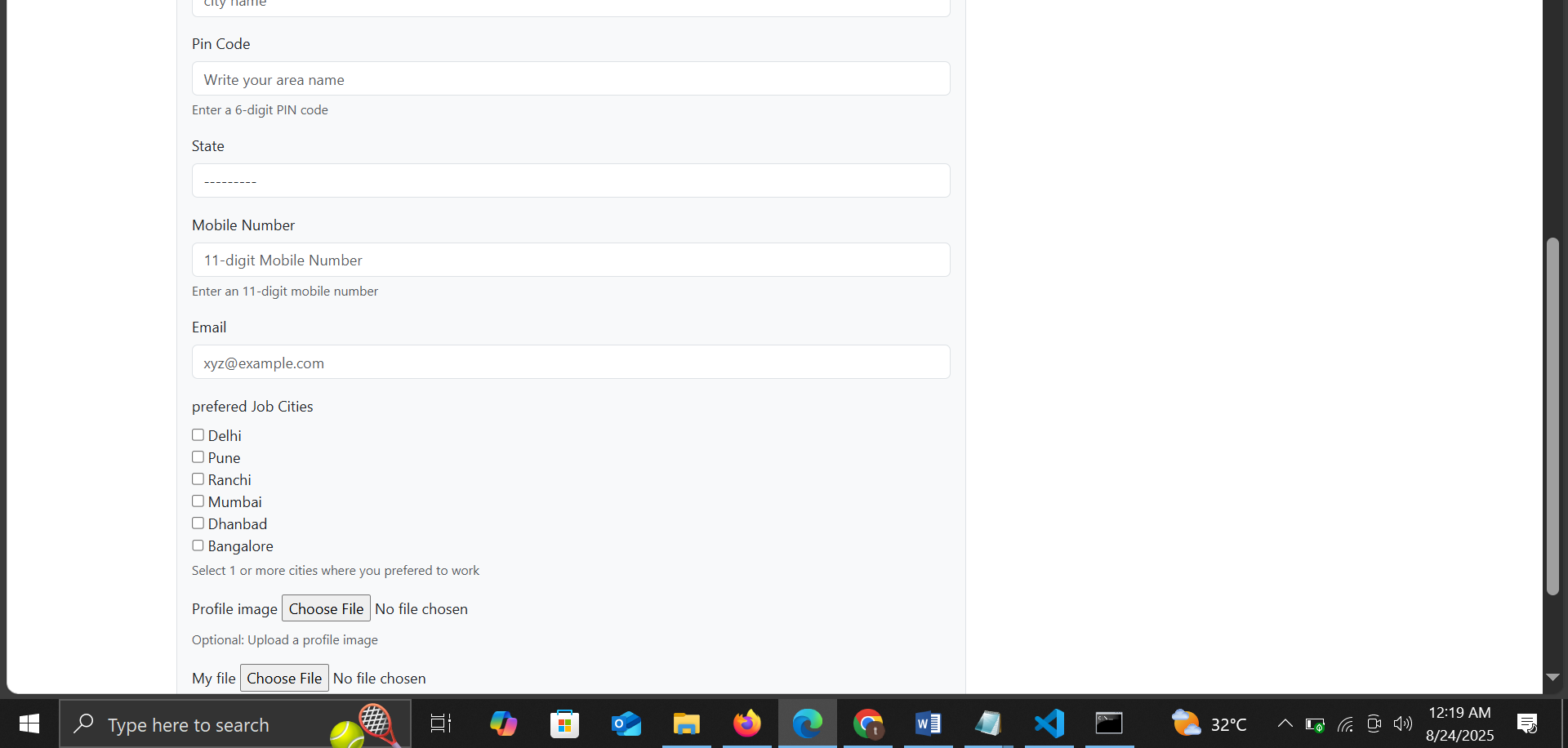
{% endfor %}

{% endblock right-content %}

Home.hhtml using for loop

Webpage look like :





One issue in that if user don’t uplaod any thing in myfile and profile image form can submit with out any error if we want user must upload both the things so use this in models.py:

from django.db import models

class Student(models.Model):

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

    my\_file = models.FileField(upload\_to="documents/", null=False, blank=False)

    profile\_image = models.ImageField(upload\_to="profiles/", null=False, blank=False)

    def \_\_str\_\_(self):

        return self.name