

Step 1: Create Author Model

```
1 class Author(models.Model):
2     user = models.OneToOneField(User, on_delete=models.CASCADE)
3     created_at = models.DateTimeField(auto_now_add=True)
4
5     def __str__(self):
6         return self.user.username
7
```

Step 2: Create RegistrationForm Form Class

```
1 # forms.py
2 from django import forms
3 from django.contrib.auth.models import User
4
5 class RegistrationForm(forms.Form):
6     username = forms.CharField(max_length=150, required=True)
7     email = forms.EmailField(required=True)
8     password = forms.CharField(widget=forms.PasswordInput, required=True)
9     confirm_password = forms.CharField(widget=forms.PasswordInput, required=True)
10
11     def clean(self):
12         cleaned_data = super().clean()
13         password = cleaned_data.get('password')
14         confirm_password = cleaned_data.get('confirm_password')
15
16         if password != confirm_password:
17             raise forms.ValidationError("Passwords do not match")
18
19         return cleaned_data
20
```

Step 3: Create User_register() View Function

```
1 # views.py
2 from django.shortcuts import render, redirect
3 from django.contrib.auth.models import User
4 from django.contrib import messages
5 from .models import Author
6 from .forms import RegistrationForm
7
8 def User_register(request):
9     if request.method == 'POST':
10         form = RegistrationForm(request.POST)
11         if form.is_valid():
12             username = form.cleaned_data['username']
13             email = form.cleaned_data['email']
14             password = form.cleaned_data['password']
15
16             # Check if username or email already exists
17             if User.objects.filter(username=username).exists():
18                 messages.error(request, "Username already taken")
19                 return render(request, 'register.html', {'form': form})
20
21             if User.objects.filter(email=email).exists():
22                 messages.error(request, "Email already registered")
23                 return render(request, 'register.html', {'form': form})
24
25             # Create User
26             user = User.objects.create_user(username=username, email=email, password=password)
27
28             # Create Author Profile
29             Author.objects.create(user=user)
30
31             messages.success(request, "Registration successful. You can now log in.")
32             return redirect('login')
33         else:
34             messages.error(request, "Invalid form submission")
35     else:
36         form = RegistrationForm()
37
38     return render(request, 'register.html', {'form': form})
39
```

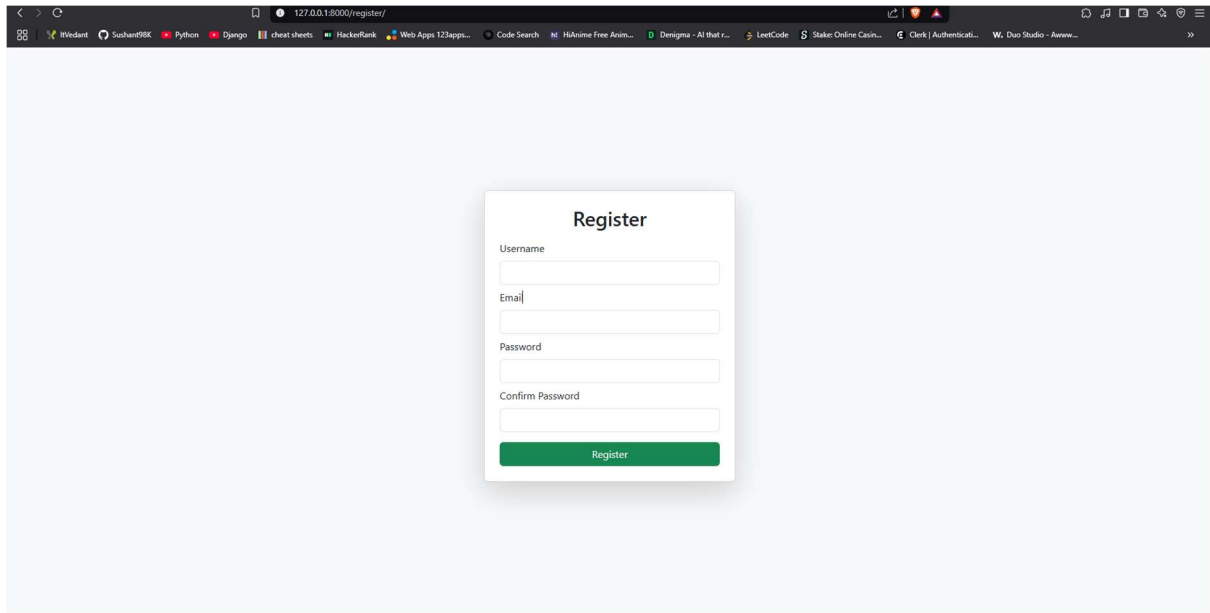
Step 4: Create register.html Template

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>User Registration</title>
5   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
6 </head>
7 <body class="d-flex vh-100 justify-content-center align-items-center bg-light">
8   <div class="card p-4 shadow-lg" style="width: 400px;">
9     <h2 class="text-center mb-3">Register</h2>
10    {% if messages %}
11      {% for message in messages %}
12        <div class="alert alert-{{ message.tags }} text-center">
13          {{ message }}
14        </div>
15      {% endfor %}
16    {% endif %}
17    <form method="POST">
18      {% csrf_token %}
19      <div class="mb-2">
20        <label for="id_username" class="form-label">Username</label>
21        <input type="text" name="username" id="id_username" class="form-control">
22      </div>
23      <div class="mb-2">
24        <label for="id_email" class="form-label">Email</label>
25        <input type="email" name="email" id="id_email" class="form-control">
26      </div>
27      <div class="mb-2">
28        <label for="id_password" class="form-label">Password</label>
29        <input type="password" name="password" id="id_password" class="form-control">
30      </div>
31      <div class="mb-3">
32        <label for="id_confirm_password" class="form-label">Confirm Password</label>
33        <input type="password" name="confirm_password" id="id_confirm_password" class="form-control">
34      </div>
35      <button type="submit" class="btn btn-success w-100">Register</button>
36    </form>
37  </div>
38 </body>
39 </html>
40
```

Step 5: Add URL Pattern

```
1 # urls.py
2 from django.urls import path
3 from . import views
4
5 urlpatterns = [
6     path('register/', views.User_register, name='register'),
7 ]
8
```

OUTPUT: -



A screenshot of a web browser displaying a registration form titled "Register". The form is centered on a light blue background. It contains four input fields: "Username", "Email", "Password", and "Confirm Password". Below these fields is a green button labeled "Register". The browser's address bar shows the URL "127.0.0.1:8000/register/". The browser's taskbar at the bottom shows various open applications, including ITVedant, Sushant98K, Python, Django, cheat sheets, HackerRank, Web Apps 123apps..., Code Search, HiAnime Free Anim..., Designa - All that r..., LeetCode, Stake Online Casin..., Clerk | Authenticat..., and Duo Studio - Awww...

Project Link:

https://github.com/Sushant98K/ITVedant/tree/main/Django/02.%20Django%20Model/Blog_Project