Create a Registration functionality for the Author/User.

Registration Form Fields

Username - text field
Email - email field
password and confirm password - password field

Validations

Before inserting data into the Model please validate

1. Data from the form field is not empty.
2. Email is in the correct format
3. Password and Confirm Password are same

Create View function -

Create a View Function named as User_register() that will add details in User Model and Author Model using same view function.

Step1 python -m venv venv

Step2 venv\Scripts\activate

Step3 pip install Django

Step4 django-admin startproject myproject

Step5 cd myproject

Step6 python manage.py startapp accounts

Step7

register.html | settings.py

myproject > myproject > settings.py > ...

```
31  # Application definition
32
33  INSTALLED_APPS = [
34      'django.contrib.admin',
35      'django.contrib.auth',
36      'django.contrib.contenttypes',
37      'django.contrib.sessions',
38      'django.contrib.messages',
39      'django.contrib.staticfiles',
40      'accounts',
41  ]
42
```

Step8

```
# accounts/models.py

from django.db import models
```

```
from django.contrib.auth.models import User

class Author(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE)
    bio = models.TextField(blank=True, null=True)

    def __str__(self):
        return self.user.username
```

Step9

python manage.py makemigrations

python manage.py migrate

Step10

```
# accounts/views.py

from django.shortcuts import render
from django.contrib.auth.models import User
from .models import Author
from django.core.validators import validate_email
from django.core.exceptions import ValidationError

def User_register(request):
    if request.method == "POST":
        username = request.POST.get('username').strip()
        email = request.POST.get('email').strip()
        password = request.POST.get('password').strip()
        confirm_password = request.POST.get('confirm_password').strip()

        # --- Validation ---
        if not username or not email or not password or not confirm_password:
            return render(request, 'register.html', {'error': 'All fields are
required.'})

        # Validate email format
        try:
            validate_email(email)
        except ValidationError:
            return render(request, 'register.html', {'error': 'Invalid email
format.'})

        # Check if passwords match
        if password != confirm_password:
            return render(request, 'register.html', {'error': 'Passwords do
not match.'})
```

```
        # Check if username or email already exists
        if User.objects.filter(username=username).exists():
            return render(request, 'register.html', {'error': 'Username
already exists.'})

        if User.objects.filter(email=email).exists():
            return render(request, 'register.html', {'error': 'Email already
registered.'})

        # --- Save to User Model ---
        user = User.objects.create_user(username=username, email=email,
password=password)

        # --- Save to Author Model ---
        author = Author.objects.create(user=user)

        return render(request, 'register.html', {'success': 'Registration
successful!'})

    return render(request, 'register.html')
```

Step11

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

urlpatterns = [
    path('register/', views.User_register, name='register'),
]
```

Step12

```
"""
URL configuration for myproject project.

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/5.2/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
```

```
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
# myproject/urls.py

from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('accounts.urls')),
]
```

Step13

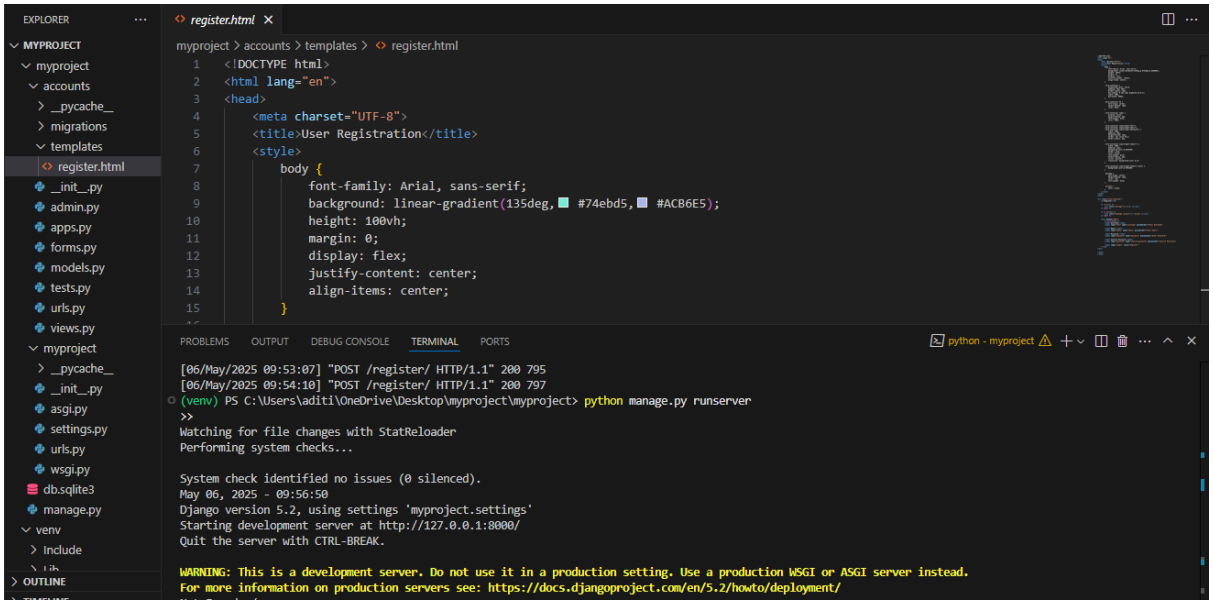
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>User Registration</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background: linear-gradient(135deg, #74ebd5, #ACB6E5);
            height: 100vh;
            margin: 0;
            display: flex;
            justify-content: center;
            align-items: center;
        }

        .form-container {
            background-color: white;
            padding: 30px 40px;
            border-radius: 12px;
            box-shadow: 0 4px 12px rgba(0,0,0,0.1);
            width: 100%;
            max-width: 400px;
        }

        .form-container h2 {
            text-align: center;
            margin-bottom: 20px;
            color: #333;
        }

        .form-container label {
            display: block;
```

```
            margin-bottom: 5px;
            font-weight: bold;
            color: #444;
        }

        .form-container input[type="text"],
        .form-container input[type="email"],
        .form-container input[type="password"] {
            width: 100%;
            padding: 10px;
            margin-bottom: 15px;
            border: 1px solid #ccc;
            border-radius: 6px;
        }

        .form-container input[type="submit"] {
            width: 100%;
            padding: 12px;
            background-color: #4CAF50;
            border: none;
            color: white;
            font-weight: bold;
            border-radius: 6px;
            cursor: pointer;
            transition: background-color 0.3s;
        }

        .form-container input[type="submit"]:hover {
            background-color: #45a049;
        }

        .message {
            text-align: center;
            margin-bottom: 15px;
            color: red;
            font-weight: bold;
        }

        .success {
            color: green;
        }
    </style>
</head>
<body>

<div class="form-container">
    <h2>Register</h2>
```

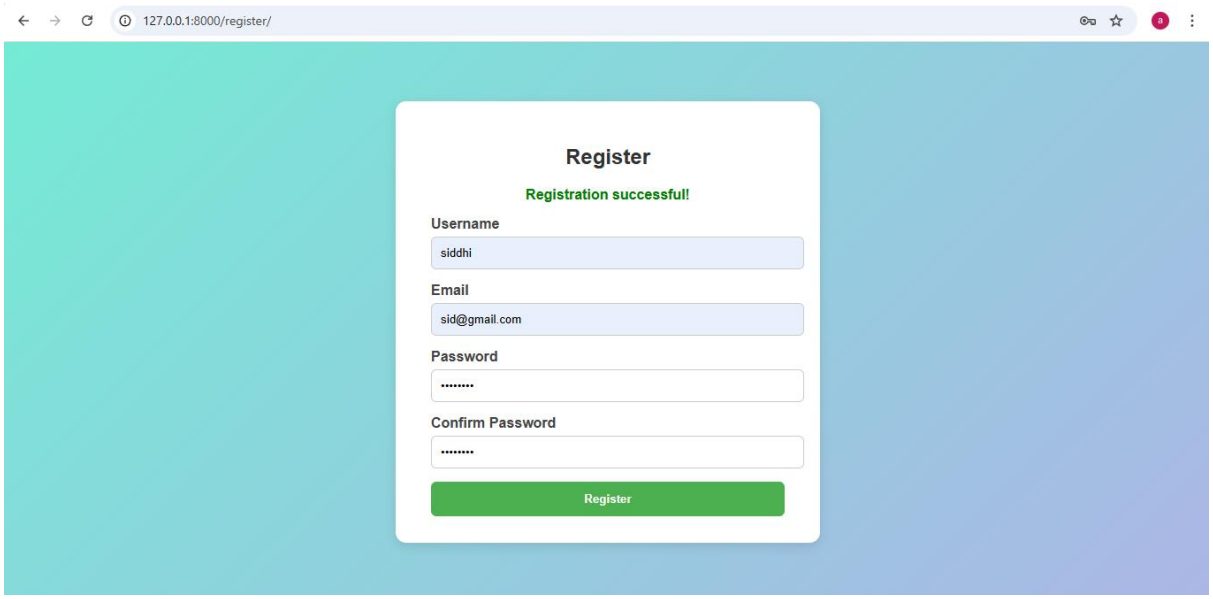
```
    {% if error %}
        <div class="message">{{ error }}</div>
    {% endif %}

    {% if success %}
        <div class="message success">{{ success }}</div>
    {% endif %}

    <form method="POST">
        {% csrf_token %}
        <label>Username</label>
        <input type="text" name="username" placeholder="Enter Username">

        <label>Email</label>
        <input type="email" name="email" placeholder="Enter Email">

        <label>Password</label>
        <input type="password" name="password" placeholder="Enter Password">

        <label>Confirm Password</label>
        <input type="password" name="confirm_password" placeholder="Confirm
Password">

        <input type="submit" value="Register">
    </form>
</div>

</body>
</html>
```

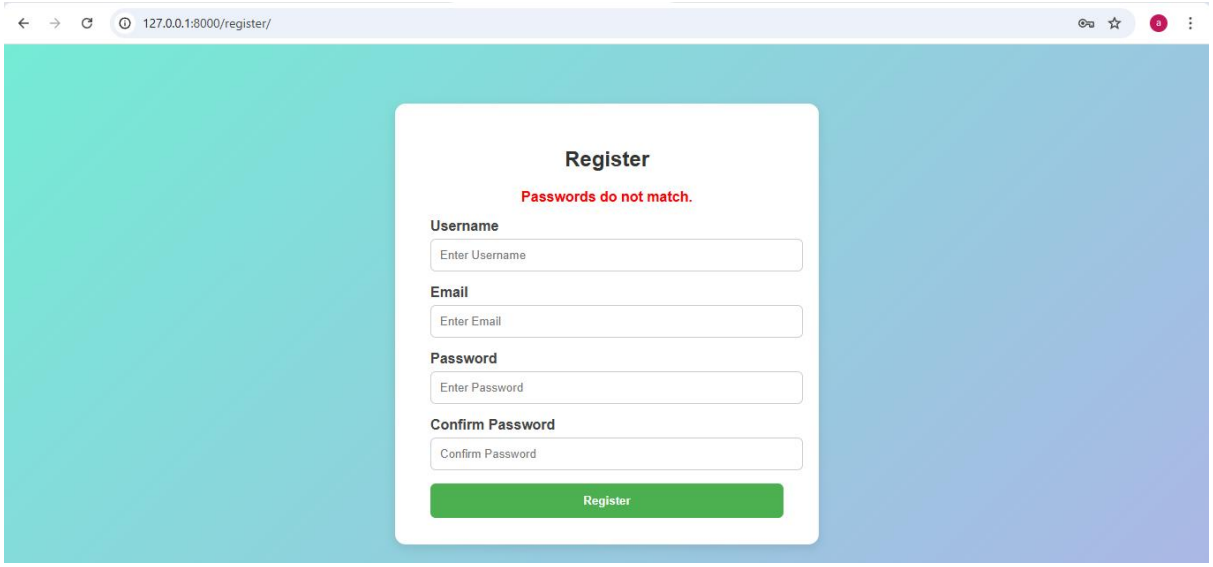
Step14

python manage.py runserver


EXPLORER
register.html
MYPROJECT
myproject
accounts
__pycache__
migrations
templates
register.html
__init__.py
admin.py
apps.py
forms.py
models.py
tests.py
urls.py
views.py
myproject
__pycache__
__init__.py
asgi.py
settings.py
urls.py
wsgi.py
db.sqlite3
manage.py
venv
Include
OUTLINE
myproject > accounts > templates > register.html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>User Registration</title>
<style>
body {
font-family: Arial, sans-serif;
background: linear-gradient(135deg, #74ebd5, #ACB6E5);
height: 100vh;
margin: 0;
display: flex;
justify-content: center;
align-items: center;
}
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
python - myproject
[06/May/2025 09:53:07] "POST /register/ HTTP/1.1" 200 795
[06/May/2025 09:54:10] "POST /register/ HTTP/1.1" 200 797
(venv) PS C:\Users\aditi\OneDrive\Desktop\myproject\myproject> python manage.py runserver
>>
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).
May 06, 2025 - 09:56:50
Django version 5.2, using settings 'myproject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead.
For more information on production servers see: https://docs.djangoproject.com/en/5.2/howto/deployment/

Step15


127.0.0.1:8000/register/
Register
Registration successful!
Username
siddhi
Email
sid@gmail.com
Password
Confirm Password
Register


127.0.0.1:8000/register/
Register
Passwords do not match.
Username
Enter Username
Email
Enter Email
Password
Enter Password
Confirm Password
Confirm Password
Register