Build a simple blog website using Django.

Step 1: Set Up the Project

1. **Install Python and Django:**
   - Make sure you have Python installed. If not, download and install it from here.
   - Install Django using pip:

```
pip install django
```

2. **Create a New Django Project:**
   - Open your terminal/command prompt.
   - Navigate to the directory where you want to create your project.
   - Run the following command to create a new Django project (replace `blog_project` with your desired project name):

```
django-admin startproject blog_project
```

3. **Navigate to the Project Directory:**

```
cd blog_project
```

4. **Create a New Django App:**
   - Run the following command to create a new app within your project (replace `blog` with your desired app name):

```
python manage.py startapp blog
```

5. **Add the App to Your Project Settings:**
   - Open `blog_project/settings.py` and add `'blog'` to the `INSTALLED_APPS` list:

python

```
INSTALLED_APPS = [
    ...
    'blog',
]
```

Step 2: Define the Blog Models

1. **Edit `models.py` in the Blog App:**
   - Open `blog/models.py` and define the `Post` model:

python

```
from django.db import models
from django.utils import timezone

class Post(models.Model):
```

```
    title = models.CharField(max_length=200)
    content = models.TextField()
    author = models.CharField(max_length=100)
    created_at = models.DateTimeField(default=timezone.now)
    updated_at = models.DateTimeField(auto_now=True)

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

2. **Create and Apply Migrations:**
   - Run the following commands to create and apply the database migrations:

```
python manage.py makemigrations
python manage.py migrate
```

Step 3: Create Views and Templates

1. **Create Views in `views.py`:**
   - Open `blog/views.py` and create views for listing posts and displaying post details:

```
python
```

```
from django.shortcuts import render, get_object_or_404
from .models import Post

def post_list(request):
    posts = Post.objects.all().order_by('-created_at')
    return render(request, 'blog/post_list.html', {'posts':
posts})

def post_detail(request, post_id):
    post = get_object_or_404(Post, id=post_id)
    return render(request, 'blog/post_detail.html', {'post':
post})
```

2. **Create URL Patterns:**
   - Open `blog/urls.py` (create it if it doesn't exist) and define the URL patterns for the blog:

```
python
```

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

urlpatterns = [
    path('', views.post_list, name='post_list'),
    path('post/<int:post_id>/', views.post_detail,
name='post_detail'),
]
```

   - Include the blog URLs in the main project's `urls.py`:

```
python
```

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

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

3. **Create Templates:**
   - Create a `templates` directory inside the `blog` app.
   - Inside the `templates` directory, create a `blog` folder.
   - Create `post_list.html` inside the `blog` folder:

```
html
<!DOCTYPE html>
<html>
<head>
    <title>Blog</title>
</head>
<body>
    <h1>Blog Posts</h1>
    <ul>
        {% for post in posts %}
            <li>
                <a href="{% url 'post_detail' post.id %}">{{
post.title }}</a> by {{ post.author }} on {{ post.created_at }}
            </li>
        {% endfor %}
    </ul>
</body>
</html>
```

   - Create `post_detail.html` inside the `blog` folder:

```
html
Copy code
<!DOCTYPE html>
<html>
<head>
    <title>{{ post.title }}</title>
</head>
<body>
    <h1>{{ post.title }}</h1>
    <p>{{ post.content }}</p>
    <p>By {{ post.author }} on {{ post.created_at }}</p>
    <a href="{% url 'post_list' %}">Back to Blog</a>
</body>
</html>
```

Step 4: Add Admin Interface

1. **Register the Post Model in Admin:**
   - Open `blog/admin.py` and register the `Post` model:

```
python
```

```
from django.contrib import admin
from .models import Post
```

```
admin.site.register(Post)
```

2. **Create a Superuser:**
   - Run the following command to create a superuser account:

```
python manage.py createsuperuser
```

3. **Run the Development Server:**
   - Start the development server:

```
python manage.py runserver
```

4. **Access the Admin Interface:**
   - Navigate to `http://127.0.0.1:8000/admin/` in your browser and log in with the superuser account.
   - You can now add, edit, and delete blog posts through the admin interface.

Step 5: Test the Blog

1. **Navigate to the Blog:**
   - Open your browser and go to `http://127.0.0.1:8000/`.
   - You should see the list of blog posts.
2. **View Post Details:**
   - Click on a post title to view its details.

By following these steps, you should have a simple blog website built with Django. You can further enhance this project by adding features like user authentication, comments, tags, and categories.