

TRAINING DAY20 REPORT

16 JULY 2025

Building a Blog Application in Django

- **Defining the Blog Model:**

```
blogapi > blog > models.py > ...
1  # blog/models.py
2  from django.db import models
3  from django.utils.text import slugify
4  from django.contrib.auth.models import User
5
6  class Post(models.Model):
7      title = models.CharField(max_length=200)
8      slug = models.SlugField(unique=True, blank=True)
9      author = models.ForeignKey(User, on_delete=models.CASCADE)
10     content = models.TextField()
11     created_at = models.DateTimeField(auto_now_add=True)
12     updated_at = models.DateTimeField(auto_now=True)
13
14     def save(self, *args, **kwargs):
15         if not self.slug:
16             self.slug = slugify(self.title)
17         super().save(*args, **kwargs)
18
19     def __str__(self):
20         return self.title
21
```

- **CharField** – short text (title of the post).
- **SlugField** – SEO-friendly URL identifier generated from the title.
- **ForeignKey** – link between Post and User (one user can have many posts).
- **auto_now_add / auto_now** – automatically handle timestamps.
- **save() override** – automatically creates a slug when saving the post.

Then run:

```
python manage.py makemigrations
```

```
python manage.py migrate
```

- **Creating Admin Interface:**

```
blogapi > blog > admin.py > ...
1  # blog/admin.py
2  from django.contrib import admin
3  from .models import Post
4
5  @admin.register(Post)
6  class PostAdmin(admin.ModelAdmin):
7      list_display = ('title', 'author', 'created_at')
8      prepopulated_fields = {'slug': ('title',)}
9
```

Now run:

```
python manage.py createsuperuser
```

Then visit /admin — you can **add, edit, or delete posts** easily.

- **Writing Views:**

The **view** processes requests and returns responses.

```

blogapi > blog > views.py > ...
1  # blog/views.py
2  from django.shortcuts import render, get_object_or_404
3  from .models import Post
4
5  def home(request):
6      posts = Post.objects.all().order_by('-created_at')
7      return render(request, 'blog/home.html', {'posts': posts})
8
9  def post_detail(request, slug):
10     post = get_object_or_404(Post, slug=slug)
11     return render(request, 'blog/post_detail.html', {'post': post})
12

```

→ home() – Fetches all posts and sends them to the homepage.

→ post_detail() – Fetches one specific post using its slug.

- **Setting Up URLs**

```

blogapi > blog > urls.py > ...
1  # blog/urls.py
2  from django.urls import path
3  from . import views
4
5  urlpatterns = [
6      path('', views.home, name='home'),
7      path('post/<slug:slug>/', views.post_detail, name='post_detail'),
8  ]
9

```

Include this inside the main project URLs:

```

18  # blogproject/urls.py
19  from django.contrib import admin
20  from django.urls import path, include
21
22  urlpatterns = [
23      path('admin/', admin.site.urls),
24      path('', include('blog.urls')),
25  ]
26
27

```

- **Templates:**

home.html

```
templates > home.html > ...
1  {% extends "base.html" %}
2
3  {% block title %}Blog Name{% endblock %}
4
5  {% block content %}
6
7      <h1>My Django Blog</h1>
8
9      <!-- Search Bar -->
10     <form method="GET" action="{% url 'home' %}">
11         <input type="text" name="q" placeholder="Search posts..." value="{{ request.GET.q }}">
12         <button type="submit">Search</button>
13     </form>
14
15     <hr>
16     {% if posts %}
17         {% for post in posts %}
18             <h2><a href="{% url 'post_detail' post.slug %}">{{ post.title }}</a></h2>
19             <p>{{ post.content|truncatewords:20 }}</p>
20             <small>By {{ post.author.username }} | {{ post.created_at|date:"M d, Y" }}</small>
21             <hr>
22         {% endfor %}
23     {% else %}
24         <p>No posts available.</p>
25     {% endif %}
26
27 {% endblock %}
28
```

post_details.html:

```
templates > post_details.html > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>{{ post.title }}</title>
5  </head>
6  <body>
7      <a href="{% url 'home' %}">Back to Home</a>
8      <h1>{{ post.title }}</h1>
9      <p>{{ post.content }}</p>
10     <small>Written by {{ post.author.username }} on {{ post.created_at|date:"M d, Y" }}</small>
11 </body>
12 </html>
13
```