

# Module – 8

## Learning Objectives

- Task – Adding Comments/Reviews to our Posts
- Task – Adding authentication to pages displayed using direct call to url
- Task – Redirecting to previously called page after performing login activity

## Task – Adding Comments/Reviews to our Posts

### blog/models.py

```
class Comment(models.Model):
    post = models.ForeignKey(Post, on_delete=models.CASCADE, related_name="comments")
    text = models.CharField(max_length=140)
    author = models.ForeignKey('accounts.CustomUser', on_delete=models.CASCADE)

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

### blog/admin.py

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

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

### blog/forms.py

```
class CommentForm(forms.ModelForm):
    text = forms.CharField(widget=forms.TextInput(), label='Comment')

    class Meta:
        model = Comment
        fields = ('text',)
```

### blog/views.py

```
def postDetailView(request, pk):
    post = get_object_or_404(Post, pk=pk)
    if request.method == "POST":
        form = CommentForm(request.POST)
        if form.is_valid():
            comment = form.save(commit=False)
            comment.author = request.user
            comment.post = post
            comment.save()
            return redirect('detailview', pk=post.pk)
    else:
        form = CommentForm()
        return render(request, 'detailview.html', {'post': post, 'form': form})
```

### blog/templates/detailview.html

```
{% extends 'base.html' %}
{% block title %}{{post.title }}{% endblock %}
{% block base%}
{% load crispy_forms_tags %}
```

```

.
.
.

<div class="card-footer">
    {% for comment in post.comments.all %}
    <p>
        <span class="font-weight-bold">{{ comment.author }}: </span>
        {{ comment }}
    </p>
    {% endfor %}
    {% if user.is_authenticated %}
    <form method="post">{% csrf_token %}
        {{ form|crispy }}
        <button type="submit" class="btn btn-sm">Submit</button>
    </form>
    {%else%}
    <a href="{% url 'login' %}">Login</a> to add comments
    {% endif%}
</div>

</div>
</div>
</div>
<br><br>
{% endblock %}

```

## Task – Adding authentication to pages displayed using direct call to url

### Blog/views.py

```

from django.shortcuts import render, get_object_or_404, redirect
from .models import Post
from .forms import PostForm, CommentForm
from django.contrib.auth.decorators import login_required

# Delete a post - authentication using basic if - else
def postDeleteView(request, pk):
    post = get_object_or_404(Post, pk=pk)
    if request.user.is_authenticated and request.user == post.author:
        post.delete()
        return redirect('listposts')
    else:
        return redirect('login')

# Form to add new object - authentication using decorators
@login_required
def postAddView(request):
    heading = 'Add Post'
    if request.method == "POST":
        form = PostForm(request.POST)
        if form.is_valid():
            post = form.save(commit=False)
            post.author = request.user

```

```
        post.save()
        return redirect('detailview', pk=post.pk)
    else:
        form = PostForm()
    return render(request, 'formview.html', {'form': form, 'heading': heading})
```

## Task – Redirecting to previously called page after performing login activity

blog/templates/detailview.html

```
{% if user.is_authenticated %}
<form method="post">{% csrf_token %}
{{ form|crispy }}
<button type="submit" class="btn btn-sm">Submit</button>
</form>
{% else %}
<a href="{% url 'login' %}?next={{request.path}}">Login</a> to add comments
{% endif %}
```