

# Module – 6

## Learning Objectives

- **Task – Creating a form to add new Post and Update Existing Post**
- Working with forms
- Add new Post, Edit Post Forms
- Adding external form styling template (crispy forms – bootstrap 4)

## blog/templates/listposts.html (link for adding new post)

```
{% extends 'base.html' %}
{% block title %}Home{% endblock %}
{% block base%}
<div class="row">
    <div class="col-md-8">
        <a href="{% url 'newpost' %}" class="btn btn-info">Add new post</a>
        <br><br>
        {% for post in posts %}
        <h3><a href="{% url 'detailview' post.pk %}">{{ post.title }}</a></h3>
        {{ post.created_date }}
        {% endfor %}
    </div>
</div>
{% endblock%}
```

## blog/templates/detailview.html (link for editing post)

```
{% extends 'base.html' %}
{% block title %}{{post.title }}{% endblock %}
{% block base%}
<div class="content container">
    <div class="row">
        <div class="card col-md-12 border-right-0 border-left-0 border-top-0 border-bottom-0">
            <div class="card-header">
                <h2>{{ post.title }}</h2>
                {{ post.created_date }}

                <span style="float: right;">
                    <a href="{% url 'editpost' post.pk %}" class="btn btn-link">Edit</a> |
                    <a href="{% url 'deletepost' post.pk %}" class="btn btn-link">Delete</a>
                </span>
            </div>
            <div>
                <p>{{ post.text }}</p>
            </div>
        </div>
    </div>
</div>
<br><br>
{% endblock %}
```

## blog/urls.py

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

urlpatterns = [
    path('', views.postListView, name='listposts'),
    path('post/<int:pk>/', views.postDetailView, name='detailview'),
    path('post/<int:pk>/delete', views.postDeleteView, name='deletepost'),
    path('post/<int:pk>/edit', views.postEditView, name='editpost'),
    path('post/new', views.postAddView, name='newpost'),
]
```

## blog/forms.py

```
from django import forms
from .models import Post

class PostForm(forms.ModelForm):
    title = forms.CharField(widget=forms.TextInput())
    text = forms.CharField(widget=forms.Textarea())

    class Meta:
        model = Post
        # fields = ('author', 'title', 'text',)
        fields = '__all__'
```

## blog/views.py

```
from django.shortcuts import render, get_object_or_404, redirect
from .models import Post
from .forms import PostForm

# Form to add new object
def postAddView(request):
    heading = 'Add Post'
    if request.method == "POST":
        form = PostForm(request.POST)
        if form.is_valid():
            post = form.save(commit=False)
            post.save()
            return redirect('detailview', pk=post.pk)
    else:
        form = PostForm()
    return render(request, 'formview.html', {'form': form, 'heading': heading})

# Form to update object
def postEditView(request, pk):
    heading = 'Edit Post'
    post = get_object_or_404(Post, pk=pk)
    if request.method == "POST":
        form = PostForm(request.POST, instance=post)
        if form.is_valid():
            post = form.save(commit=False)
            post.save()
            return redirect('detailview', pk=post.pk)
    else:
        form = PostForm(instance=post)
    return render(request, 'formview.html', {'form': form, 'heading': heading})
```

## blog/templates/formview.html/

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

```

<div class="row">
  <div class="col-md-12" align="centre">
    <form method="post">
      <div class="card col-md-8">
        <div class="card-header bg-primary" align="centre"><h3 align="center">{{heading}}</h3></div>
        <div class="card-body" align="left">
          {% csrf_token %}
          {{ form|crispy }}
        </div>
        <div class="card-footer" align="center">
          <button type="submit" class="btn btn-success">Save</button>
        </div>
      </div>
    </form>
    <br><br>
  </div>
</div>
{% endblock%}

```

The CSRF middleware and template tag provides easy-to-use protection against Cross Site Request Forgeries. This type of attack occurs when a malicious website contains a link, a form button or some JavaScript that is intended to perform some action on your website, using the credentials of a logged-in user who visits the malicious site in their browser. A related type of attack, ‘login CSRF’, where an attacking site tricks a user’s browser into logging into a site with someone else’s credentials, is also covered.

Settings required to load crispy forms

MyBlog/settings.py

```

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'blog',
    'accounts',
    'crispy_forms',
]
.
.
.
CRISPY_TEMPLATE_PACK = 'bootstrap4'

```