Module – 7

Learning Objectives

- **Task – Adding Login/Logout and Signup Facility**
- Login and signup Forms
- Using existing auth.User model and UserCreationForm
- **Task – Allowing only the logged in users to add new post**
- Authentication Functionality
- **Task – Allowing only the owner of the post to edit/delete the post**
- Authorisation Functionality
- **Task – Adding additional gender, dob and age field to existing user model**
- Creating a custom user model
- **Task – Password change and reset functionality**

Login/Logout Functionality

templates/base.html

```
{% extends 'scripts.html' %}
{% load static %}
{% block scripts_content %}
<div class="content container">
    <div class="page-header">
        <img src="{% static 'media/headerimg.png' %}" width="1110" height="150"/>
        <div class="progress" style="height: 2px">
            <div class="progress-bar bg-warning" style="width:100%; height: 5px"></div>
        </div>
        <nav class="navbar navbar-expand bg-dark navbar-dark">
            <div class="container justify-content-md-start">
                <ul class="navbar-nav">
                    <a href="{% url 'listposts' %}"> Home</a>
                    &nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp;
                    <a href="{% url 'aboutpage' %}"> About us</a>
                    &nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp;
                    <a href="{% url 'contactpage' %}"> Contact us</a>
                </ul>
            </div>
            <div class="container justify-content-md-end">
                <ul class="navbar-nav">
                  {% if user.is_authenticated %}
                  <a href="{% url 'logout' %}">logout</a>
                  {% else %}
                  <a href="{% url 'login' %}">Login</a>
                  {% endif %}
                </ul>
            </div>
        </nav>
    </div>
    <div class="progress" style="height: 2px">
        <div class="progress-bar bg-warning" style="width:100%"></div>
    </div>
    <br>
    {% block base%}
    {% endblock%}
    <div class="fixed-bottom" align="center">
        <div class="progress" style="height: 2px">
            <div class="progress-bar bg-warning" style="width:100%; height: 5px"></div>
        </div>
        <div class="bg-dark">
            Copyright Django@PetaaBytes
        </div>
    </div>
</div>
{% endblock %}
```

MyBlog/urls.py

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('about/', TemplateView.as_view(template_name="about.html"), name='aboutpage'),
    path('contact/', TemplateView.as_view(template_name="contact.html"), name='contactpage'),
    path('', include('blog.urls')),
    path('accounts/', include('django.contrib.auth.urls')),
    path('accounts/', include('accounts.urls')),
]
```

templates/registration/login.html (url and view for login in taken care by Django)

(Place this file in project level templates/registration folder)

```
{% extends 'scripts.html' %}
{% block title %}Login{% endblock %}
{% block scripts_content %}
{% load crispy_forms_tags %}
<div class="content container" align="center">
    <form method="post">
        <div class="card col-md-6">
            <div class="card-header bg-primary"><h3>Login</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">Login</button>
                <br>
                Not a member yet?? <a href="{% url 'signup'%}">Register here.</a>
            </div>
        </div>
    </form>
</div>

{% endblock %}
```

MyBlog/settings.py

```
LOGIN_REDIRECT_URL = 'listposts'
LOGOUT_REDIRECT_URL = 'listposts'
```

Sign-Up Functionality

accounts/urls.py

```
from django.urls import path
from . import views
urlpatterns = [
    path('signup/', views.SignUpView.as_view(), name='signup'),
]
```

accounts/views.py

```
from django.contrib.auth.forms import UserCreationForm
from django.urls import reverse_lazy
from django.views.generic import CreateView

# Class – based view for sign-up
class SignUpView(CreateView):
    form_class = UserCreationForm
    success_url = reverse_lazy('login')
    template_name = 'signup.html'

    def form_valid(self, form):
        return super().form_valid(form)
```

- reverse() function: It's similar to the url template tag which use to convert namespaced url to real url pattern.
- reverse_lazy() function: This is a reverse() function's lazy version. It prevents error to occur when URLConf is not loaded. Generally, we use this function in case below:
    - providing a reversed URL as the url attribute of a generic class-based view.
    - providing a reversed URL to a decorator (such as the login_url argument for the django.contrib.auth.decorators.permission_required()decorator)
    - providing a reversed URL as a default value for a parameter in a function's signature.

accounts/templates/signup.html

```
{% extends 'scripts.html' %}
{% block title %}Sign Up{% endblock %}
{% block scripts_content %}
{% load crispy_forms_tags %}
<div class="content container" align="center">
    <form method="post">
        <div class="card col-md-6">
            <div class="card-header bg-primary"><h3>Sign up</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">Register</button>
                <br>
                Already a member?? <a href="{% url 'login'%}">Login here.</a>
            </div>
        </div>
    </form>
</div>

{% endblock %}
```

Customise User creation Form

accounts/forms.py

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

class CustomUserCreationForm(UserCreationForm):
    # Fileds from auth.User model
    password1 = forms.CharField(widget=forms.PasswordInput(), label='Password')
    password2 = forms.CharField(widget=forms.PasswordInput(), label='Confirm Password')
    first_name = forms.CharField(widget=forms.TextInput())
    last_name = forms.CharField(widget=forms.TextInput())
    email = forms.CharField(widget=forms.EmailInput())

    class Meta(UserCreationForm):
        model = User
        fields = ('first_name', 'last_name', 'email', 'username', 'password1', 'password2',)
        help_texts = {
            'username': None,
        }
```

Authentication and Authorisation

blog/templates/listposts.html

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

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
        # Removed the author field
        fields = ('title', 'text',)
```

blog/views.py

```
# Form to add new object
def postAddView(request):
    if request.user.is_authenticated:
        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})
    else:
        return redirect('login')
```

Creating Custom User Model

Note: Whenever creating a custom model start as a fresh project or delete the original database and all migrations. Make sure migrations are applied first (using makemigrations) and then migrate command is to be excuted.

accounts/models.py

```
from django.contrib.auth.models import AbstractUser
from django.db import models
from datetime import date

class CustomUser(AbstractUser):
    gender = models.CharField(max_length=10)
    dob = models.DateField(blank=True, null=True)
    age = models.PositiveIntegerField(default=0)

    def calAge(self):
        d = date.today()
        self.age = d.year - self.dob.year
        self.save()
```

accounts/forms.py

```
from django import forms
from django.contrib.auth.forms import UserCreationForm
from .models import CustomUser

class CustomUserCreationForm(UserCreationForm):

    # Fileds from auth.User model
    password1 = forms.CharField(widget=forms.PasswordInput(), label='Password')
    password2 = forms.CharField(widget=forms.PasswordInput(), label='Confirm Password')

    first_name = forms.CharField(widget=forms.TextInput())
    last_name = forms.CharField(widget=forms.TextInput())
    email = forms.CharField(widget=forms.EmailInput())

    gender = forms.ChoiceField(choices=(('male', 'Male'),('female', 'Female')), widget=forms.RadioSelect())

    class Meta(UserCreationForm):
        model = CustomUser
        fields = ('first_name', 'last_name', 'dob', 'gender','email', 'username', 'password1', 'password2',)
        help_texts = {
            'username': None,
        }
```

accounts/admin.py

```
from django.contrib import admin
from django.contrib.auth.admin import UserAdmin
from .models import CustomUser

admin.site.register(CustomUser)
```

MyBlog/settings.py

```
AUTH_USER_MODEL = 'accounts.CustomUser'
```

Change Password Functionality

templates/base.html

```
{% extends 'scripts.html' %}
{% load static %}
{% block scripts_content %}
<div class="content container">
    <div class="page-header">
.
.
.
            <div class="container justify-content-md-end">
                <ul class="navbar-nav">
                  {% if user.is_authenticated %}
                        <li class="nav-item dropdown">
                        <a class="nav-link dropdown-toggle" href="#" id="navbardrop" data-toggle="dropdown">
Hello {{user.username}}</a>
                        <div class="dropdown-menu">
                          <a class="dropdown-item" href="{% url 'password_change' %}">Change Password</a>
                          <a class="dropdown-item" href="{% url 'logout' %}">logout</a>
                        </div>
                        </li>
                  {% else %}
                  <a href="{% url 'login' %}">Login</a>
                  {% endif %}
                </ul>
            </div>
.
.
.
```

templates/registration/password_change_form.html (url and view for login in taken care by Django but can be changed using UserChangeForm)

(Place this file in project level templates/registration folder)

```
{% extends 'scripts.html' %}
{% block title %}Change Password{% endblock %}
{% block scripts_content %}
{% load crispy_forms_tags %}
<div class="content container" align="center">
    <form method="post">
        <div class="card col-md-6">
            <div class="card-header bg-primary"><h3>Password Change</h3></div>
            <div class="card-body" align="left">
              <p>Please enter your old password, for security's sake, and then enter your new password twice so we can verify you typed it in correctly.</p>
              {% csrf_token %}
              {{ form|crispy }}
          </div>
          <div class="card-footer" align="center">
              <button type="submit" class="btn btn-success">Change Password</button>
              <br>
              Don't want to change password?? <a href="{% url 'listposts' %}">Home</a>
          </div>
        </div>
    </form>
</div>

{% endblock %}
```

templates/registration/password_change_done.html

(Place this file in project level templates/registration folder)

```
{% extends 'scripts.html' %}
{% block title %}Change Password{% endblock %}
{% block scripts_content %}
<div class="content container" align="center">
        <div class="card col-md-6">
            <div class="card-header bg-primary"><h3>Password Change Successfull</h3></div>
          <h1>Password change successful. Click here to <a href="{% url 'login'%}">Login</a></h1>
          </div>
</div>
{% endblock %}
```

Password Reset Functionality – Using email

SendGrid, MailGun are some of the popular services for sending transactional that can be used by Django.

templates/registration/login.html

```
{% extends 'scripts.html' %}
{% block title %}Login{% endblock %}
{% block scripts_content %}
{% load crispy_forms_tags %}
<div class="content container" align="center">
    <form method="post">
        <div class="card col-md-6">
            <div class="card-header bg-primary"><h3>Login</h3></div>
            <div class="card-body" align="left">
                {% csrf_token %}
                {{ form|crispy }}
                # urls.py is already configured to handle all the user accounts related requests
                 Forgot Password?? <a href="{% url 'password_reset' %}">Reset here.</a>
            </div>
            <div class="card-footer" align="center">
                <button type="submit" class="btn btn-success">Login</button>
                <br>
                Not a member yet?? <a href="{% url 'signup'%}">Register here.</a>
            </div>
        </div>
    </form>
</div>
{% endblock %}
```

MyBlog/settings.py

```
# Add this line to bottom of the file

EMAIL_BACKEND = 'django.core.mail.backends.console.EmailBackend'
```

templates/registration/password_reset_form.html (Place this file in project level templates/registration folder)

```
{% extends 'scripts.html' %}
{% block title %}Password Reset{% endblock %}
{% block scripts_content %}
{% load crispy_forms_tags %}
<div class="content container" align="center">
    <form method="post">
        <div class="card col-md-6">
            <div class="card-header bg-primary"><h3>Password Reset</h3></div>
            <div class="card-body" align="left">
                <p>Enter your email address below, and we'll email instructions for setting a new one.</p>
                {% csrf_token %}
                {{ form|crispy }}
            </div>
            <div class="card-footer" align="center">
                <button type="submit" class="btn btn-success">Send me Instructions</button>
                <br>
                Don't want to reset password?? <a href="{% url 'login' %}">Login</a>
            </div>
        </div>
    </form>
</div>
{% endblock %}
```

templates/registration/password_reset_done.html (Place this file in project level templates/registration folder)

```
{% extends 'scripts.html' %}
{% block title %}Password Reset{% endblock %}
{% block scripts_content %}

<div class="content container" align="center">
        <div class="card col-md-6">
            <div class="card-header bg-primary"><h3>Check your inbox.</h3></div>
            <div class="card-body" align="left">
                <p>We've emailed you instructions for setting your password. You should receive the email 
shortly!</p>
            </div>
            <div class="card-footer" align="center">
                Don't want to reset password?? <a href="{% url 'login' %}">Login</a>
            </div>
        </div>
</div>
{% endblock %}
```

templates/registration/password_reset_confirm.html (Place this file in project level templates/registration folder)

```
{% extends 'scripts.html' %}
{% block title %}Password Reset{% endblock %}
{% block scripts_content %}
{% load crispy_forms_tags %}
<div class="content container" align="center">
    <form method="post">
        <div class="card col-md-6">
            <div class="card-header bg-primary"><h3>Set New Password</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">Change Password</button>
            </div>
        </div>
    </form>
</div>
{% endblock %}
```

templates/registration/password_reset_complete.html (Place this file in project level templates/registration folder)

```
{% extends 'scripts.html' %}
{% block title %}Password Reset{% endblock %}
{% block scripts_content %}
<div class="content container" align="center">
    <div class="card col-md-6">
```

```
        <div class="card-header bg-primary"><h3>Password Reset Complete</h3></div>
        <div class="card-body" align="left">
            <p>Your new password has been set. <a href="{% url 'login' %}">Click here</a> to login.
            </p>
        </div>
    </div>
</div>
{% endblock %}
```