|  |  |
| --- | --- |
| **1** | **python manage.py startapp accounts** |
| 2 | We will create a project-level URL for the accounts path in /moviereviews/  urls.py:  urlpatterns = [  path('admin/', admin.site.urls),  path('', movieViews.home, name='home'),  path('about/', movieViews.about, name='about'),  path('signup/', movieViews.signup, name='signup'),  path('news/', include('news.urls')),  path('movie/', include('movie.urls')),  **path('accounts/', include('accounts.urls')), ]** |
| 3 | We will put all the paths related to the accounts app (for example, signup, login, and logout) in its own urls.py – that is, /accounts/urls.py (create this file with the following code):  from django.urls import path  from . import views  urlpatterns = [  path('signupaccount/', views.signupaccount, name='signupaccount'),  ] |
| 4 | Next, create def signupaccount in /accounts/views.py with the following code in bold:  **from django.contrib.auth.forms import UserCreationForm**  **def signupaccount(request):**  **return render(request, 'signupaccount.html',**  **{'form':UserCreationForm})** |
| 5 | Next, create /accounts/templates/signupaccount.html and simply fill in the following:  {{ form }} |
| 6 | Replace the entire /accounts/templates/signupaccount.html with the following:  {% extends 'base.html' %}  {% block content %}  <div class="card mb-3">  <div class="row g-0">  <div>  <div class="card-body">  <h5 class="card-title">Sign Up</h5>  <p class="card-text">  <form method="POST">  {% csrf\_token %}  {{ form.as\_p }}  <button type="submit"  class="btn btn-primary">  Sign Up  </button>  </form>  </p>  </div>  </div>  </div>  </div>  {% endblock content %} |
| 7 | Add the following in bold in /accounts/views.py:  from django.shortcuts import render  from django.contrib.auth.forms import UserCreationForm  from django.contrib.auth.models import User  from django.contrib.auth import login  from django.shortcuts import redirect  # Create your views here.  def signupaccount(request):  if request.method == 'GET':  return render(request, 'signupaccount.html',{'form':UserCreationForm})  else:  if request.POST['password1'] == request.POST['password2']:  user = User.objects.create\_user(  request.POST['username'],  password= request.POST['password1'])  user.save()  login(request, user)  return redirect('home') |
| 8 | What happens if password1 doesn't match password2? To handle such an error, we add the following else block in bold in /accounts/views.py:  def signupaccount(request):  if request.method == 'GET':  return render(request, 'signupaccount.html',  {'form':UserCreationForm})  else:  if request.POST['password1'] ==  request.POST['password2']:  user = User.objects.create\_user(  request.POST['username'],  password=request.POST['password1'])  user.save()  login(request, user)  return redirect('home')  **else:**  **return render(request, 'signupaccount.html',**  **{'form':UserCreationForm,**  **'error':'Passwords do not match'})** |
| 9 | In /accounts/templates/signupaccount.html, add the following code in bold:  {% extends 'base.html' %}  {% block content %}  <div class="card mb-3">  <div class="row g-0">  <div>  <div class="card-body">  <h5 class="card-title">Sign Up</h5>  **{% if error %}**  **<div class="alert alert-danger mt-3" role="alert">**  **{{ error }}**  **</div>**  **{% endif %}**  <p class="card-text"> |
| 10 | Checking if a username already exists. here (include this code in /accounts/views.py):  from django.shortcuts import redirect  **from django.db import IntegrityError**  def signupaccount(request):  if request.method == 'GET':  return render(request, 'signupaccount.html',  {'form':UserCreationForm})  else:  if request.POST['password1'] ==  request.POST['password2']:  **try:**  user = User.objects.create\_user(  request.POST['username'],  password=request.POST['password1'])  user.save()  login(request, user)  return redirect('home')  **except IntegrityError:**  **return render(request,'signupaccount.html',**  **{'form':UserCreationForm,**  **'error':'Username already taken. Choose**  **New username.'})**  else:  return render(request, 'signupaccount.html',  {'form':UserCreationForm,  'error':'Passwords do not match'}) |
| 11 | In /accounts/, create a new file called forms.py and fill it in with the following:  from django.contrib.auth.forms import UserCreationForm  class UserCreateForm(UserCreationForm):  def \_\_init\_\_(self, \*args, \*\*kwargs):  super(UserCreateForm, self).\_\_init\_\_(\*args, \*\*kwargs)  for fieldname in ['username', 'password1', 'password2']:  self.fields[fieldname].help\_text = None  self.fields[fieldname].widget.attrs.update({'class': 'form-control'}) |
|  |  |
| 12 | **Using UserCreateForm**  Finally, in /accounts/views.py, change the following in bold to use  UserCreateForm instead of UserCreationForm:  from django.shortcuts import render  **from django.contrib.auth.forms import UserCreationForm**  **from .forms import UserCreateForm**  from django.contrib.auth.models import User  …  def signupaccount(request):  if request.method == 'GET':  return render(request, 'signupaccount.html',  {'form':**UserCreateForm**})  else:  if request.POST['password1'] ==  request.POST['password2']:  try:  user = User.objects.create\_user(  request.POST['username'],  password=request.POST['password1'])  user.save()  login(request, user)  return redirect('home')  except IntegrityError:  return render(request,  'signupaccount.html',  {'form':**UserCreateForm**,  'error':'Username already taken. Choose  new username.'})  else:  return render(request, 'signupaccount.html',  {'form':**UserCreateForm**,  'error':'Passwords do not match'}) |
| 13 | In moviereviews/templates/base.html, add the following code in bold:  <nav class="navbar navbar-expand-lg navbar-light  bg-light mb-3">  …  <div class="collapse navbar-collapse"  id="navbarNavAltMarkup">  <div class="navbar-nav ms-auto">  <a class="nav-link" href="{% url 'news' %}">  News</a>  **{% if user.is\_authenticated %}**  **<a class="nav-link" href="#">**  **Logout ({{ user.username }})**  **</a>**  **{% else %}**  <a class="nav-link" href="#">Login</a>  <a class="nav-link" href="#">Sign Up</a>  **{% endif %}**  </div>  </div> |
| 14 | Create the logout path in /accounts/urls.py with the following in bold:  urlpatterns = [  path('signupaccount/', views.signupaccount, name='signupaccount'),  **path('logout/', views.logoutaccount, name='logoutaccount'),**  ] |
| 15 | In /accounts/views.py, implement the logoutaccount function with the  following in bold:  from django.contrib.auth.models import User  from django.contrib.auth import login**, logout**  from django.shortcuts import redirect  from django.db import IntegrityError  …  **def logoutaccount(request):**  **logout(request)**  **return redirect('home')** |
| 16 | We then need to have <a href> in the logout button call the logout path. We also have the signup button call the signupaccount path. So, in moviereviews/templates/base.html, add the following in bold:  <nav class="navbar navbar-expand-lg navbar-light  bg-light mb-3">  <div class="collapse navbar-collapse"  id="navbarNavAltMarkup">  <div class="navbar-nav ms-auto">  <a class="nav-link" href="{% url 'news' %}">  News</a>  {% if user.is\_authenticated %}  <a class="nav-link"  **href="{% url 'logoutaccount' %}">**  Logout ({{ user.username }})  </a>  {% else %}  <a class="nav-link" href="#">Login</a>  <a class="nav-link"  href**="{% url'signupaccount' %}">**  Sign Up  </a>  {% endif %}  </div>  </div> |
| 17 | Create a login path in /accounts/urls.py:  urlpatterns = [  path('signupaccount/', views.signupaccount, name='signupaccount'),  path('logout/', views.logoutaccount, name='logoutaccount'),  **path('login/', views.loginaccount, name='loginaccount'),**  ] |
| 18 | In /accounts/views.py, implement loginaccount with the following  in bold:  from django.contrib.auth.models import User  **from django.contrib.auth.forms import AuthenticationForm**  from django.contrib.auth import login, logout**,**  **authenticate**  ...  **def loginaccount(request):**  **if request.method == 'GET':**  **return render(request, 'loginaccount.html', {'form':AuthenticationForm})**  **else:**  **user = authenticate(request, username=request.POST['username'],**  **password=request.POST['password'])**  **if user is None:**  **return render(request,'loginaccount.html',{'form': AuthenticationForm(),**  **'error': 'username and password do not match'})**  **else:**  **login(request,user)**  **return redirect('home')** |
| 19 | Create the new file, accounts/templates/loginaccount.html, copy the  markup from accounts/templates/signupaccount.html, and change the  labeling from 'Sign Up' to 'Login':  {% extends 'base.html' %}  {% block content %}  <div class="card mb-3">  <div class="row g-0">  <div>  <div class="card-body">  <h5 class="card-title">**Login**</h5>  {% if error %}  <div class="alert alert-danger mt-3"  role="alert">  {{ error }}  </div>  {% endif %}  <p class="card-text">  <form method="POST">  {% csrf\_token %}  {{ form.as\_p }}  <button type="submit"  class="btn btn-primary">  **Login**  </button>  </form>  </p>  </div>  </div>  </div>  </div>  {% endblock content %} |
| 20 | Finally, in moviereviews/templates/base.html, we set href for  loginaccount:  {% if user.is\_authenticated %}  <a class="nav-link" href="{% url  'logoutaccount' %}">  Logout ({{ user.username }})  </a>  {% else %}  <a class="nav-link"  href="**{% url 'loginaccount' %}**">  Login  </a>  <a class="nav-link"  href="{% url 'signupaccount' %}">  Sign Up  </a>  {% endif %} |

|  |  |
| --- | --- |
| 21 | In /movie/views.py, add def createreview:  from django.shortcuts import get\_object\_or\_404**, redirect**  from .models import Movie**, Review**  **from .forms import ReviewForm**  …  **def createreview(request, movie\_id):**  **movie = get\_object\_or\_404(Movie,pk=movie\_id)**  **if request.method == 'GET':**  **return render(request, 'createreview.html',**  **{'form':ReviewForm(), 'movie': movie})**  **else:**  **try:**  **form = ReviewForm(request.POST)**  **newReview = form.save(commit=False)**  **newReview.user = request.user**  **newReview.movie = movie**  **newReview.save()**  **return redirect('detail',newReview.movie.id)**  **except ValueError:**  **return render(request,'createreview.html',**  **{'form':ReviewForm(),'error':'bad data passed in'})** |
| 22 | In /movie/templates, create a new file, createreview.html, with the  following code:  {% extends 'base.html' %}  {% block content %}  <div class="card mb-3">  <div class="row g-0">  <div>  <div class="card-body">  <h5 class="card-title">Add Review for {{movie.title }}</h5>  {% if error %}  <div class="alert alert-danger mt-3" role="alert">  {{ error }}  </div>  {% endif %}  <p class="card-text">  <form method="POST">  {% csrf\_token %}  {{ form.as\_p }}  <button type="submit"  class="btn btn-primary">  Add Review  </button>  </form>  </p>  </div>  </div>  </div>  </div>  {% endblock content %} |
| 23 | To create the review form, we can make use of ModelForm provided by Django to automatically create forms from models. In /movie, create the forms.py file and fill it with the following code:  from django.forms import ModelForm, Textarea  from .models import Review  class ReviewForm(ModelForm):  def \_\_init\_\_(self, \*args, \*\*kwargs):  super(ModelForm, self).\_\_init\_\_(\*args,  \*\*kwargs)  self.fields['text'].widget.attrs.update(  {'class': 'form-control'})  self.fields['watchAgain'].widget.attrs.update(  {'class': 'form-check-input'})  class Meta:  model = Review  fields = ['text','watchAgain']  labels = {  'watchAgain': ('Watch Again')  }  widgets = {  'text': Textarea(attrs={'rows': 4}),  } |
|  |  |
|  |  |