**SOURCE CODE**

New User Register:

{% extends 'recommend/base.html'%}

{% block title %}Register{% endblock %}

{% block body %}

<div class="container-fluid">

<div class="row">

<div class="col-sm-10 col-md-6 col-md-offset-3">

<div class="panel panel-default">

<div class="panel-body">

<h3>Register for an Account</h3>

{% if error\_message %}

<p><strong>{{ error\_message }}</strong></p>

{% endif %}

<form class="form-horizontal" role="form" action="" method="post" enctype="multipart/form-data">

{% csrf\_token %}

{% include 'recommend/form\_template.html' %}

<div class="form-group">

<div class="col-sm-offset-2 col-sm-10">

<button type="submit" class="btn btn-success">Submit</button>

</div>

</div>

</form>

</div>

<div class="panel-footer">

Already have an account? <a href="{% url 'login' %}">Click here</a> to log in.

</div>

</div>

</div>

</div>

</div>

{% endblock %}

Login:

{% extends 'recommend/base.html'%}

{% block title %}Log In{% endblock %}

{% block body %}

<div class="container-fluid">

<div class="row">

<div class="col-sm-12 col-md-6 col-md-offset-3">

<div class="panel panel-default">

<div class="panel-body">

<h3>Log In</h3>

{% if error\_message %}

<p><strong>{{ error\_message }}</strong></p>

{% endif %}

<form class="form-horizontal" role="form" action="#" method="post" enctype="multipart/form-data">

{% csrf\_token %}

<div class="form-group">

<label class="control-label col-sm-2" for="id\_username">

Username:

</label>

<div class="col-sm-10">

<input id="id\_username" maxlength="30" name="username" type="text">

</div>

</div>

<div class="form-group">

<label class="control-label col-sm-2" for="id\_password">

Password:

</label>

<div class="col-sm-10">

<input id="id\_password" maxlength="30" name="password" type="password">

</div>

</div>

<div class="form-group">

<div class="col-sm-offset-2 col-sm-10">

<button type="submit" class="btn btn-success">Submit</button>

</div>

</div>

</form>

</div>

<div class="panel-footer">

Don't have an account? <a href="{% url 'signup' %}">Click here</a> to register.

</div>

</div>

</div>

</div>

</div>

{% endblock %}

List Page:

{% extends 'recommend/base.html'%}

{% block body %}

{% if not request.user.is\_authenticated %}

<br>

{% else %}

<center style="color:#fff"><h2>Welcome {{user.username}}!!</h2></center>

{% endif %}

<div class="container" >

<div class="col-xs-6 col-xs-offset-3">

<form class="form-horizontal" role="search" method="get" action=''>

<div class="form-group">

<div class="input-group">

<input type="text" class="form-control input-lg " name="q" value="{{ request.GET.q }}" placeholder= "Search Movies "/>

<span class='input-group-btn'>

<button class='button1' type='submit'>Search</button>

</span>

</div>

</div>

</form>

</div>

</div>

<p style="text-align:left;">

<span style="float:right;">

<a href="{% url 'recommend' %}" class="btn"><button style="color:#fff;background-color:#4CAF50;padding:10px">Get Movie Recommendation</button></a>

<h2>Movie List</h2>

</span>

</p>

<hr>

<br>

<!-- Movie list -->

<div class="row" >

{% if movies %}

{% for movie in movies %}

<div class="col-sm-2 col-md-2">

<div class="thumbnail" style="border-radius: 22px 22px 22px 22px;">

<center><h5>{{movie.title}}</h5></center>

<a href="{% url 'detail' movie.id %}">

<img src="{{ movie.movie\_logo.url}}" class="img-responsive" style="border-radius: 22px 22px 22px 22px;">

</a>

<!-- <h5>{{movie.genre}}</h5> -->

</div>

</div>

{% endfor %}

{% endif %}

</div>

</div>

{% endblock %}

Recommendation Model:

{% extends 'recommend/base.html'%}

{% block title %}Recommendations{% endblock %}

{% block body %}

<div class="container-fluid">

<h2>Recommendations for you</h2>

</div>

<hr>

<br>

<div class="container-fluid">

<!-- Movie list -->

<div class="row">

{% if movie\_list %}

{% for movie in movie\_list %}

<div class="col-sm-2 col-md-2 ">

<div class="thumbnail" style="border-radius: 22px 22px 22px 22px;">

<center><h4>{{movie.title}}</h4></center>

<a href="{% url 'detail' movie.id %}">

<img src="{{ movie.movie\_logo.url}}" class="img-responsive" style="border-radius: 22px 22px 22px 22px;">

</a>

<!-- <center><h5>{{movie.genre}}</h5></center>-->

</div>

</div>

{% endfor %}

{% endif %}

</div>

</div>

{% endblock %}

View Model:

from django.contrib.auth import authenticate, login

from django.contrib.auth import logout

from django.shortcuts import render, get\_object\_or\_404, redirect

from .forms import \*

from django.http import Http404

from .models import Movie, Myrating, MyList

from django.db.models import Q

from django.contrib import messages

from django.http import HttpResponseRedirect

from django.db.models import Case, When

import pandas as pd

# Create your views here.

def index(request):

movies = Movie.objects.all()

query = request.GET.get('q')

if query:

movies = Movie.objects.filter(Q(title\_\_icontains=query)).distinct()

return render(request, 'recommend/list.html', {'movies': movies})

return render(request, 'recommend/list.html', {'movies': movies})

# Show details of the movie

def detail(request, movie\_id):

if not request.user.is\_authenticated:

return redirect("login")

if not request.user.is\_active:

raise Http404

movies = get\_object\_or\_404(Movie, id=movie\_id)

movie = Movie.objects.get(id=movie\_id)

temp = list(MyList.objects.all().values().filter(movie\_id=movie\_id,user=request.user))

if temp:

update = temp[0]['watch']

else:

update = False

if request.method == "POST":

# For my list

if 'watch' in request.POST:

watch\_flag = request.POST['watch']

if watch\_flag == 'on':

update = True

else:

update = False

if MyList.objects.all().values().filter(movie\_id=movie\_id,user=request.user):

MyList.objects.all().values().filter(movie\_id=movie\_id,user=request.user).update(watch=update)

else:

q=MyList(user=request.user,movie=movie,watch=update)

q.save()

if update:

messages.success(request, "Movie added to your list!")

else:

messages.success(request, "Movie removed from your list!")

# For rating

else:

rate = request.POST['rating']

if Myrating.objects.all().values().filter(movie\_id=movie\_id,user=request.user):

Myrating.objects.all().values().filter(movie\_id=movie\_id,user=request.user).update(rating=rate)

else:

q=Myrating(user=request.user,movie=movie,rating=rate)

q.save()

messages.success(request, "Rating has been submitted!")

return HttpResponseRedirect(request.META.get('HTTP\_REFERER'))

out = list(Myrating.objects.filter(user=request.user.id).values())

# To display ratings in the movie detail page

movie\_rating = 0

rate\_flag = False

for each in out:

if each['movie\_id'] == movie\_id:

movie\_rating = each['rating']

rate\_flag = True

break

context = {'movies': movies,'movie\_rating':movie\_rating,'rate\_flag':rate\_flag,'update':update}

return render(request, 'recommend/detail.html', context)

# MyList functionality

def watch(request):

if not request.user.is\_authenticated:

return redirect("login")

if not request.user.is\_active:

raise Http404

movies = Movie.objects.filter(mylist\_\_watch=True,mylist\_\_user=request.user)

query = request.GET.get('q')

if query:

movies = Movie.objects.filter(Q(title\_\_icontains=query)).distinct()

return render(request, 'recommend/watch.html', {'movies': movies})

return render(request, 'recommend/watch.html', {'movies': movies})

# To get similar movies based on user rating

def get\_similar(movie\_name,rating,corrMatrix):

similar\_ratings = corrMatrix[movie\_name]\*(rating-2.5)

similar\_ratings = similar\_ratings.sort\_values(ascending=False)

return similar\_ratings

# Recommendation Algorithm

def recommend(request):

if not request.user.is\_authenticated:

return redirect("login")

if not request.user.is\_active:

raise Http404

movie\_rating=pd.DataFrame(list(Myrating.objects.all().values()))

new\_user=movie\_rating.user\_id.unique().shape[0]

current\_user\_id= request.user.id

# if new user not rated any movie

if current\_user\_id>new\_user:

movie=Movie.objects.get(id=19)

q=Myrating(user=request.user,movie=movie,rating=0)

q.save()

userRatings = movie\_rating.pivot\_table(index=['user\_id'],columns=['movie\_id'],values='rating')

userRatings = userRatings.fillna(0,axis=1)

corrMatrix = userRatings.corr(method='pearson')

user = pd.DataFrame(list(Myrating.objects.filter(user=request.user).values())).drop(['user\_id','id'],axis=1)

user\_filtered = [tuple(x) for x in user.values]

movie\_id\_watched = [each[0] for each in user\_filtered]

similar\_movies = pd.DataFrame()

for movie,rating in user\_filtered:

similar\_movies = similar\_movies.append(get\_similar(movie,rating,corrMatrix),ignore\_index = True)

movies\_id = list(similar\_movies.sum().sort\_values(ascending=False).index)

movies\_id\_recommend = [each for each in movies\_id if each not in movie\_id\_watched]

preserved = Case(\*[When(pk=pk, then=pos) for pos, pk in enumerate(movies\_id\_recommend)])

movie\_list=list(Movie.objects.filter(id\_\_in = movies\_id\_recommend).order\_by(preserved)[:10])

context = {'movie\_list': movie\_list}

return render(request, 'recommend/recommend.html', context)

# Register user

def signUp(request):

form = UserForm(request.POST or None)

if form.is\_valid():

user = form.save(commit=False)

username = form.cleaned\_data['username']

password = form.cleaned\_data['password']

user.set\_password(password)

user.save()

user = authenticate(username=username, password=password)

if user is not None:

if user.is\_active:

login(request, user)

return redirect("index")

context = {'form': form}

return render(request, 'recommend/signUp.html', context)

# Login User

def Login(request):

if request.method == "POST":

username = request.POST['username']

password = request.POST['password']

user = authenticate(username=username, password=password)

if user is not None:

if user.is\_active:

login(request, user)

return redirect("index")

else:

return render(request, 'recommend/login.html', {'error\_message': 'Your account disable'})

else:

return render(request, 'recommend/login.html', {'error\_message': 'Invalid Login'})

return render(request, 'recommend/login.html')

# Logout user

def Logout(request):

logout(request)

return redirect("login")