**SAMPLE CODE:**

**User side views:**

from ast import alias

from concurrent.futures import process

from django.shortcuts import render

# Create your views here.

from django.shortcuts import render, HttpResponse

from django.contrib import messages

import twitter

from .forms import UserRegistrationForm

from .models import UserRegistrationModel

from django.conf import settings

import pandas as pd

# Create your views here.

def UserRegisterActions(request):

if request.method == 'POST':

form = UserRegistrationForm(request.POST)

if form.is\_valid():

print('Data is Valid')

form.save()

messages.success(request, 'You have been successfully registered')

form = UserRegistrationForm()

return render(request, 'UserRegistrations.html', {'form': form})

else:

messages.success(request, 'Email or Mobile Already Existed')

print("Invalid form")

else:

form = UserRegistrationForm()

return render(request, 'UserRegistrations.html', {'form': form})

def UserLoginCheck(request):

if request.method == "POST":

loginid = request.POST.get('loginid')

pswd = request.POST.get('pswd')

print("Login ID = ", loginid, ' Password = ', pswd)

try:

check = UserRegistrationModel.objects.get(

loginid=loginid, password=pswd)

status = check.status

print('Status is = ', status)

if status == "activated":

request.session['id'] = check.id

request.session['loggeduser'] = check.name

request.session['loginid'] = loginid

request.session['email'] = check.email

print("User id At", check.id, status)

return render(request, 'users/UserHomePage.html', {})

else:

messages.success(request, 'Your Account Not at activated')

return render(request, 'UserLogin.html')

except Exception as e:

print('Exception is ', str(e))

pass

messages.success(request, 'Invalid Login id and password')

return render(request, 'UserLogin.html', {})

def UserHome(request):

return render(request, 'users/UserHomePage.html', {})

def DatasetView(request):

path = settings.MEDIA\_ROOT + "//" + 'twitter.csv'

df = pd.read\_csv(path, nrows=100)

df = df.to\_html

return render(request, 'users/viewdataset.html', {'data': df})

def usrtwitterFNDML(request):

from .utility import twitterMLEDA

svm\_acc, svm\_report = twitterMLEDA.process\_SVM()

svm\_report = pd.DataFrame(svm\_report).transpose()

svm\_report = pd.DataFrame(svm\_report)

nb\_acc, nb\_report = twitterMLEDA.process\_naiveBayes()

nb\_report = pd.DataFrame(nb\_report).transpose()

nb\_report = pd.DataFrame(nb\_report)

return render(request, 'users/twitterMl.html',

{

'svm\_report': svm\_report.to\_html, 'svm\_acc': svm\_acc,

'nb\_report': nb\_report.to\_html, 'nb\_acc': nb\_acc,

})

def predictTrustWorthy(request):

if request.method == 'POST':

test\_user\_data = request.POST.get('news')

print(test\_user\_data)

from .utility import twitterMLEDA

result = twitterMLEDA.fake\_news\_det(test\_user\_data)

return render(request, 'users/testform.html', {'msg': result})

else:

return render(request, 'users/testform.html', {})

**Base.html**

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Detecting A Twitter Cyberbullying Using Machine Learning</title>

<meta content="width=device-width, initial-scale=1.0" name="viewport">

<meta content="Free Website Template" name="keywords">

<meta content="Free Website Template" name="description">

<link href="{% static 'img/favicon.ico' %}" rel="icon">

<link href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@300;400;600;700;800&display=swap" rel="stylesheet">

<link href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" rel="stylesheet">

<link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.10.0/css/all.min.css" rel="stylesheet">

<link href="{% static 'lib/animate/animate.min.css' %}" rel="stylesheet">

<link href="{% static 'lib/owlcarousel/owl.carousel.min.css' %}" rel="stylesheet">

<link href="{% static 'lib/lightbox/css/lightbox.min.css' %}" rel="stylesheet">

<link href="{% static 'css/style.css' %}" rel="stylesheet">

</head>

<body>

<div class=" navbar navbar-expand-lg bg-dark navbar-dark ">

<div class="container-fluid">

<a href="index.html" class="navbar-brand"> <span style="color:rgb(172, 37, 42);">Detecting A Twitter</span></a>

<button type="button" class="navbar-toggler" data-toggle="collapse" data-target="#navbarCollapse">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse justify-content-between" id="navbarCollapse">

<div class="navbar-nav ml-auto">

<a href="{% url 'index' %}" class="nav-item nav-link" style="color:rgba(210, 62, 9, 0.733);">Home</a>

<a href="{% url 'UserLogin' %}" class="nav-item nav-link" style="color:rgba(68, 16, 200, 0.844);">User</a>

<a href="{% url 'AdminLogin' %}" class="nav-item nav-link" style="color:rgb(204, 11, 11);">Admin</a>

<a href="{% url 'UserRegister' %}" class="nav-item nav-link" style="color:rgb(233, 23, 191);">Registration</a>

</div>

</div>

</div>

</div>

{%block contents%}

{%endblock%}

<!-- <div class="container copyright">

<div class="row">

<div class="col-md-6">

</div>

<div class="col-md-6">

</div>

</div>

</div>

</div> -->

<!-- Footer End -->

<!-- <a href="#" class="back-to-top"><i class="fa fa-chevron-up"></i></a> -->

<!-- JavaScript Libraries -->

<!-- <script src="https://code.jquery.com/jquery-3.4.1.min.js"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.bundle.min.js"></script>

<script src="{% static 'lib/easing/easing.min.js' %}"></script>

<script src="{% static 'lib/owlcarousel/owl.carousel.min.js' %}"></script>

<script src="{% static 'lib/isotope/isotope.pkgd.min.js' %}"></script>

<script src="{% static 'lib/lightbox/js/lightbox.min.js' %}"></script> -->

<!-- Contact Javascript File -->

<!-- <script src="{% static 'mail/jqBootstrapValidation.min.js' %}"></script>

<script src="{% static 'mail/contact.js' %}"></script> -->

<!-- Template Javascript -->

<!-- <script src="{% static 'js/main.js' %}"></script> -->

</body>

</html>

**Admin side views:**

from django.shortcuts import render

from django.contrib import messages

from users.forms import UserRegistrationForm

from users.models import UserRegistrationModel

# Create your views here.

def AdminLoginCheck(request):

if request.method == 'POST':

usrid = request.POST.get('loginid')

pswd = request.POST.get('pswd')

print("User ID is = ", usrid)

if usrid == 'admin' and pswd == 'admin':

return render(request, 'admins/AdminHome.html')

else:

messages.success(request, 'Please Check Your Login Details')

return render(request, 'AdminLogin.html', {})

def AdminHome(request):

return render(request, 'admins/AdminHome.html',{})

def RegisterUsersView(request):

data = UserRegistrationModel.objects.all()

return render(request,'admins/viewregisterusers.html',{'data':data})

def ActivaUsers(request):

if request.method == 'GET':

id = request.GET.get('uid')

status = 'activated'

print("PID = ", id, status)

UserRegistrationModel.objects.filter(id=id).update(status=status)

data = UserRegistrationModel.objects.all()

return render(request,'admins/viewregisterusers.html',{'data':data})