**SOURCE CODE**

**Admins side views:**

from django.shortcuts import render, HttpResponse

from django.contrib import messages

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})

**Base.html:**

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Prediction of Abnormal Human Behaviour Using Machine Learning</title>

<meta content="" name="description">

<meta content="" name="keywords">

<!-- Favicons -->

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

<link href="{% static 'img/apple-touch-icon.png' %}" rel="apple-touch-icon">

<!-- Google Fonts -->

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=EB+Garamond:wght@400;500&family=Inter:wght@400;500&family=Playfair+Display:ital,wght@0,400;0,700;1,400;1,700&display=swap" rel="stylesheet">

<!-- Vendor CSS Files -->

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

<link href="{% static 'vendor/bootstrap-icons/bootstrap-icons.css' %}" rel="stylesheet">

<link href="{% static 'vendor/swiper/swiper-bundle.min.css' %}" rel="stylesheet">

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

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

<!-- Template Main CSS Files -->

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

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

<!-- =======================================================

\* Template Name: ZenBlog - v1.3.0

\* Template URL: https://bootstrapmade.com/zenblog-bootstrap-blog-template/

\* Author: BootstrapMade.com

\* License: https:///bootstrapmade.com/license/

======================================================== -->

</head>

<body>

<!-- ======= Header ======= -->

<header id="header" class="header d-flex align-items-center fixed-top">

<div class="container-fluid container-xl d-flex align-items-center justify-content-between">

<a href="index.html" class="logo d-flex align-items-center">

<!-- Uncomment the line below if you also wish to use an image logo -->

<!-- <img src="assets/img/logo.png" alt=""> -->

<h1>Prediction of Abnormal Human Behaviour Using Machine Learning</h1>

</a>

<nav id="navbar" class="navbar">

<ul>

<li class=""><a href="{% url 'index' %}">Home</a></li>

<li class=""><a href="{% url 'UserLogin' %}">EDA TESTER</a></li>

<li class=""><a href="{% url 'AdminLogin' %}">EDA MODELLER</a></li>

<li class=""><a href="{% url 'UserRegister' %}"> GET AUTORIZATION</a></li>

</ul>

</ul>

</nav><!-- .navbar -->

</div>

</header><!-- End Header -->

<main id="main">

{%block contents%}

{%endblock%}

</main>

<a href="#" class="scroll-top d-flex align-items-center justify-content-center"><i class="bi bi-arrow-up-short"></i></a>

<!-- Vendor JS Files -->

<script src="{% static 'vendor/bootstrap/js/bootstrap.bundle.min.js' %}"></script>

<script src="{% static 'vendor/swiper/swiper-bundle.min.js' %}"></script>

<script src="{% static 'vendor/glightbox/js/glightbox.min.js' %}"></script>

<script src="{% static 'vendor/aos/aos.js' %}"></script>

<script src="{% static 'vendor/php-email-form/validate.js' %}"></script>

<!-- Template Main JS File -->

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

</body>

</html>

**Users side views:**

# Create your views here.

from django.shortcuts import render, HttpResponse

from django.contrib import messages

from .forms import UserRegistrationForm

from .models import UserRegistrationModel

from django.conf import settings

from django.core.files.storage import FileSystemStorage

# 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):

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

def UserClassification(request):

import pandas as pd

from .utility import Abnormal\_Classification

rf\_report = Abnormal\_Classification.process\_randomForest()

dt\_report = Abnormal\_Classification.process\_decesionTree()

knn\_report = Abnormal\_Classification.process\_knn()

ksvm\_report = Abnormal\_Classification.process\_ksvm()

lsvm\_report = Abnormal\_Classification.process\_lsvm()

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

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

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

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

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

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

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

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

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

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

# report\_df.to\_csv("rf\_report.csv")

return render(request, 'users/cl\_reports.html',

{'rf': rf\_report.to\_html, 'dt': dt\_report.to\_html, 'knn': knn\_report.to\_html,

'ksvm': ksvm\_report.to\_html, 'lsvm': lsvm\_report.to\_html})

def UserPredictions(request):

if request.method == 'POST':

image\_file = request.FILES['file']

# let's check if it is a csv file

# if not image\_file.name.endswith('.png'):

# messages.error(request, 'THIS IS NOT A PNG FILE')

fs = FileSystemStorage(location="media/endusertest/")

filename = fs.save(image\_file.name, image\_file)

# detect\_filename = fs.save(image\_file.name, image\_file)

uploaded\_file\_url = "/media/endusertest/" + filename # fs.url(filename)

print("Image path ", uploaded\_file\_url)

from .utility.predections import predict\_user\_input

result = predict\_user\_input(filename)

print("Result=", result)

return render(request, "users/UploadForm.html", {'path': uploaded\_file\_url, 'result': result})

else:

return render(request, "users/UploadForm.html", {})