**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>Automated Drug Resistance Prediction of Tuberculosis</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 href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i|Krub:300,300i,400,400i,500,500i,600,600i,700,700i|Poppins:300,300i,400,400i,500,500i,600,600i,700,700i"

rel="stylesheet">

<!-- Vendor CSS Files -->

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

<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/boxicons/css/boxicons.min.css' %}" rel="stylesheet">

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

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

<!-- Template Main CSS File -->

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

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

\* Template Name: Bikin - v4.10.0

\* Template URL: https://bootstrapmade.com/bikin-free-simple-landing-page-template/

\* Author: BootstrapMade.com

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

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

</head>

<body>

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

<header id="header" class="fixed-top">

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

<h1 class="logo"><a href="#">Drug Resistance and Sensitivity</a></h1>

<!-- Uncomment below if you prefer to use an image logo -->

<!-- <a href="index.html" class="logo"><img src="assets/img/logo.png" alt="" class="img-fluid"></a>-->

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

<ul>

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

<li><a href="{%url 'UserLogin'%}" class="scroll-link">User</a></li>

<li><a href="{%url 'AdminLogin'%}" class="scroll-link">Admin</a></li>

<li><a href="{%url 'UserRegister'%}" class="scroll-link" >Regitser</a></li>

</ul>

<i class="bi bi-list mobile-nav-toggle"></i>

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

</div>

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

<!-- ======= Hero Section ======= -->

<section id="hero" class="d-flex align-items-center">

{% block contents %}

{% endblock %}

</section><!-- End Hero -->

<main id="main">

</main><!-- End #main -->

<!-- Vendor JS Files -->

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

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

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

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

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

**Views.py:**

from django.shortcuts import render, HttpResponse

from django.contrib import messages

from .forms import UserRegistrationForm

from .models import UserRegistrationModel

from django.core.files.storage import FileSystemStorage

from matplotlib import pyplot as plt

# 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('loginname')

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/UserHome.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/UserHome.html', {})

def TuberculosisTraining(request):

from .utility.StartTraining import tuberculosis\_training\_start

history, val\_acc, val\_loss,metrcis = tuberculosis\_training\_start()

mylist = zip(history.get('loss'), history.get('accuracy'),history.get('val\_loss'),history.get('val\_accuracy'),history.get('lr'))

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

{'history': mylist, 'val\_acc': val\_acc, 'val\_loss': val\_loss,'metrcis': metrcis})

def ConventionalTraining(request):

from .utility.ConventionalApproach import tuberculosis\_training\_start

history, val\_acc, val\_loss, metrcis = tuberculosis\_training\_start()

mylist = zip(history.get('loss'), history.get('accuracy'), history.get('val\_loss'), history.get('val\_accuracy'),

history.get('lr'))

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

{'history': mylist, 'val\_acc': val\_acc, 'val\_loss': val\_loss, 'metrcis': metrcis})

def TestMyModel(request):

if request.method == 'POST':

myfile = request.FILES['file']

fs = FileSystemStorage()

filename = fs.save(myfile.name, myfile)

uploaded\_file\_url = fs.url(filename)

from .utility import TestDRDS\_Model

result = TestDRDS\_Model.start\_test(filename)

print('Result:', result)

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

else:

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