**SOURCE CODE**

**User side views.py**

from django.conf import settings

from django.shortcuts import render

# from django.http import HttpResponse

from .forms import UserRegistrationForm

from django.contrib import messages

from .models import UserRegistrationModel

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 successfully registered')

form = UserRegistrationForm()

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

else:

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

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('password')

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 has not been activated by Admin.')

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 ml(request):

if request.method == 'POST':

temp\_high = int(request.POST.get('temp\_high'))

temp\_avg = int(request.POST.get('temp\_avg'))

temp\_low = int(request.POST.get('temp\_low'))

dew\_point\_high = int(request.POST.get('dew\_point\_high'))

dew\_point\_avg = int(request.POST.get('dew\_point\_avg'))

dew\_point\_low = int(request.POST.get('dew\_point\_low'))

humidity\_high = int(request.POST.get('humidity\_high'))

humidity\_avg = int(request.POST.get('humidity\_avg'))

humidity\_low = int(request.POST.get('humidity\_low'))

sea\_level\_pressure\_avg\_inches = float(request.POST.get('sea\_level\_pressure\_avg\_inches'))

visibility\_high = int(request.POST.get('visibility\_high'))

visibility\_avg = int(request.POST.get('visibility\_avg'))

visibility\_low = int(request.POST.get('visibility\_low'))

wind\_high = int(request.POST.get('wind\_high'))

wind\_avg = int(request.POST.get('wind\_avg'))

wind\_gust = int(request.POST.get('wind\_gust'))

# print(temp\_high, temp\_low, temp\_avg, dew\_point\_high, dew\_point\_low, dew\_point\_avg,

# humidity\_low, humidity\_avg, humidity\_high, sea\_level\_pressure\_avg\_inches,

# visibility\_avg, visibility\_low, visibility\_high, wind\_avg, wind\_high, wind\_gust, sep='\n')

#

from .utility.ml import do\_prediction

result = do\_prediction(

temp\_high=temp\_high,

temp\_avg=temp\_avg,

temp\_low=temp\_low,

dew\_point\_high=dew\_point\_high,

dew\_point\_avg=dew\_point\_avg,

dew\_point\_low=dew\_point\_low,

humidity\_high=humidity\_high,

humidity\_avg=humidity\_avg,

humidity\_low=humidity\_low,

sea\_level\_pressure\_avg\_inches=sea\_level\_pressure\_avg\_inches,

visibility\_high=visibility\_high,

visibility\_avg=visibility\_avg,

visibility\_low=visibility\_low,

wind\_high=wind\_high,

wind\_avg=wind\_avg,

wind\_gust=wind\_gust,

)

return render(request, 'users/ml.html', {'result': f"The precipitation in inches for the input is:{result}"})

# return render(request, 'users/prediction.html')

else:

# return render(request, 'users/prediction.html')

return render(request, 'users/ml.html')

def dataset(request):

dataset\_url = settings.DATASET\_URL

data = pd.read\_csv(dataset\_url)

context = {

'data': data.to\_html(

index=False,

classes=['table table-striped table-bordered table-hover table-sm']

).replace('<tr style="text-align: right;">', '<tr>')

}

return render(request, 'users/view\_data.html', context)

def ann(request):

from .utility import artificial\_neural\_network as ann

print('Lets print the results... ')

regressor = ann.build\_regressor()

# Evaluate Loss (Mean Squared Error), Mean Absolute Error, Accuracy,

regressor\_results = regressor.evaluate(ann.X\_test, ann.y\_test)

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Regressor Result \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

loss = regressor\_results[0]

mae = regressor\_results[1]

accuracy = regressor\_results[2]

print('\_\_LOSS\_\_:', loss)

print('\_\_MAE\_\_:', mae)

print('\_\_ACCURACY\_\_:', accuracy)

context = {

'loss': loss,

'mae': mae,

'accuracy': accuracy

}

return render(request, 'users/ann.html', context)

def mlr(request):

from .utility.ml import mae\_mse\_r2\_score

result = mae\_mse\_r2\_score()

context = {

'mean\_absolute\_error': result[0],

'mean\_squared\_error': result[1],

'r2\_score': result[2]

}

return render(request, 'users/mlr.html', context)

**index.html:**

{% extends 'base.html' %}

{% block content %}

{% load static %}

<!-- Hero section -->

<head>

<meta charset="UTF-8"/>

<meta http-equiv="X-UA-Compatible" content="IE=edge"/>

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

<!-- Bootstrap CSS -->

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

<!-- Web page CSS -->

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

<!-- Simple lightbox CSS -->

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

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

<!-- Favicons -->

<link

rel="apple-touch-icon"

sizes="180x180"

href="{% static 'assets/icons/apple-touch-icon.png' %}"

/>

<link

rel="icon"

type="image/png"

sizes="32x32"

href="{% static 'assets/icons/favicon-32x32.png' %}"

/>

<link

rel="icon"

type="image/png"

sizes="16x16"

href="{% static 'assets/icons/favicon-16x16.png' %}"

/>

<title>🌧 </title>

</head>

<section class="hero">

<div class="intro-text">

<h1>

<span class="hear"> Rainfall Prediction using </span> <br/>

<span class="connecting"> Multiple linear Regression Model</span>

</h1>

{# <p>#}

{# Rainfall Prediction using Multiple Linear Regressions Model <br/>#}

{##}

{# </p>#}

<a class="button-1 text-white" href="{% url 'learn\_more' %}">Learn More</a>

<a class="button-2 text-white" href="{% url 'UserRegister' %}">Register</a>

</div>

<div class="i-frame">

<iframe width="560" height="315" src="https://www.youtube.com/embed/ruRwEPpEfCg" title="YouTube video player"

frameborder="0"

allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture"

allowfullscreen></iframe>

<div class="stand-1"></div>

<div class="stand-2"></div>

</div>

</section>

{% endblock %}

**Base.html:**

<!DOCTYPE html>

{% load static %}

<html lang="en">

<head>

<meta charset="UTF-8"/>

<meta http-equiv="X-UA-Compatible" content="IE=edge"/>

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

<!-- Bootstrap CSS -->

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

<!-- Web page CSS -->

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

<!-- Simple lightbox CSS -->

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

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

<!-- Favicons -->

<link

rel="apple-touch-icon"

sizes="180x180"

href="{% static 'assets/icons/apple-touch-icon.png' %}"

/>

<link

rel="icon"

type="image/png"

sizes="32x32"

href="{% static 'assets/icons/favicon-32x32.png' %}"

/>

<link

rel="icon"

type="image/png"

sizes="16x16"

href="{% static 'assets/icons/favicon-16x16.png' %}"

/>

<title>🌧 </title>

</head>

<body>

<nav>

<a href="/" class="logo">

<h1>

<span class="jab"> 🌧</span>

<span class="fist"></span>

<span class="tv">Prediction</span>

</h1>

</a>

<!-- Dark/light theme switcher -->

<div class="theme-switch">

<input type="checkbox" class="checkbox" id="checkbox"/>

<label for="checkbox" class="label">

<ion-icon name="partly-sunny-outline" class="sun"></ion-icon>

<ion-icon name="moon-outline" class="moon"></ion-icon>

<div class="switcher"></div>

</label>

</div>

<ul>

<li class="nav-item">

<a href="{% url 'AdminLogin' %}"

class="nav-link {% if request.path == '/AdminLogin/' %} active {% endif %} "

id="nav-link">Admin</a>

</li>

<li class="nav-item">

<a href="{% url 'UserLogin' %}" class="nav-link {% if request.path == '/UserLogin/' %} active {% endif %}"

id="nav-link">Users</a>

</li>

<li class="nav-item">

<a href="{% url 'UserRegister' %}"

class="nav-link {% if request.path == '/UserRegister/' %} active {% endif %} button-1 "

id="nav-link">

Register

</a>

</li>

</ul>

<!-- Bars -->

<div class="hamburger" id="hamburger">

<span class="bar"></span>

<span class="bar"></span>

<span class="bar"></span>

</div>

</nav>

{% block content %}

{% endblock %}

<footer>&copy;2022. All Rights Reserverd</footer>

<!-- Scroll to top button -->

<i class="scroll-up" id="scroll-up">

<img src="{% static 'assets/icons/icons8-upward-arrow.png' %}"

class="socicon up-arrow"

alt="up-arrow"

/>

</i>

<!-- Web page script -->

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

<!-- Ion icons CDN -->

{#<script type="module" src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.esm.js"></script>#}

<script type="module" src="{% static 'ionicons/ionicons.esm.js' %}"></script>

<!-- Bootstrap JS -->

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

<!-- Simple lightbox -->

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

<script>

var lightbox = new SimpleLightbox(".stars-gallery a");

</script>

</body>

</html>