
Source Code

User side views.py

```
# 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
import os
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 viewDataSet(request):
    path = os.path.join(settings.MEDIA_ROOT, 'FinalDataSet.csv')
    df = pd.read_csv(path, encoding="ISO-8859-1")
    df = df[['source', 'user_name', 'FinalLabel', 'text', 'user_followers_count']]
    # print(df.head())
    df = df.to_html(index=None)
    return render(request, 'users/viewTweetDataset.html', {'data': df})
```

```
def userMachineLearning(request):
    from .utility import mlprocessings
    cr_lg = mlprocessings.start_logisticRegression()
    cr_nb = mlprocessings.start_naivebayes()
    cr_svm = mlprocessings.start_svm()
    cr_rnn = mlprocessings.start_recurrentNeurals()
    return render(request, 'users/MIResults.html', {'cr_lg': cr_lg, 'cr_nb': cr_nb, 'cr_svm':
cr_svm, 'cr_rnn': cr_rnn})

def start_test_predictions(request):
    if request.method == 'POST':
        tweets = request.POST.get('tweets')
        import os
        import pandas as pd
        from django.conf import settings
        from sklearn.feature_extraction.text import CountVectorizer
        import pickle
        from sklearn.model_selection import train_test_split
        path1 = os.path.join(settings.MEDIA_ROOT, 'FinalDataSet.csv')
        path = os.path.join(settings.MEDIA_ROOT, 'fakenews.alex')
        df = pd.read_csv(path1, encoding="ISO-8859-1")
        df.shape

        df['FinalLabel'] = df.FinalLabel.map({'REAL': 1, 'FAKE': 0})
        X_train, X_test, y_train, y_test = train_test_split(df['text'],
                                                            df['FinalLabel'],
                                                            random_state=42)

        # Instantiate the CountVectorizer method
        count_vector = CountVectorizer(stop_words='english', lowercase=True)
        training_data = count_vector.fit_transform(X_train)

        # Transform testing data and return the matrix. Note we are not fitting the testing
        data into the CountVectorizer()
```

```

testing_data = count_vector.transform(X_test)
test = count_vector.transform([tweets])
model = pickle.load(open(path, 'rb'))
pred = model.predict(test)
if pred[0] == 1:
    msg = 'REAL'
else:
    msg = 'FAKE'
print("====>", pred)
return render(request, 'users/test_pred.html', {'tweet': tweets, 'msg': msg})
else:
    return render(request, 'users/test_pred.html', {})

```

Base.html:

```

{% load static %}
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>fake news</title>
    <meta name="description" content="Free Bootstrap Theme by BootstrapMade.com">
    <meta name="keywords" content="free website templates, free bootstrap themes, free
template, free bootstrap, free website template">
    <link rel="stylesheet" type="text/css"
href="https://fonts.googleapis.com/css?family=Open+Sans|Raleway|Candal">
    <link rel="stylesheet" type="text/css" href="{ % static 'css/font-awesome.min.css'
% } ">
    <link rel="stylesheet" type="text/css" href="{ % static 'css/bootstrap.min.css' % } ">
    <link rel="stylesheet" type="text/css" href="{ % static 'css/style.css' % } ">

```

```

</head>
<body id="myPage" data-spy="scroll" data-target=".navbar" data-offset="60">
  <section id="banner" class="banner">
    <div class="bg-color">
      <nav class="navbar navbar-default navbar-fixed-top">
        <div class="container">
          <div class="col-md-12">
            <div class="navbar-header">
              <button type="button" class="navbar-toggle" data-toggle="collapse" data-
target="#myNavbar">
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
              </button>
            </div>

            <div class="collapse navbar-collapse navbar-right" id="myNavbar">
              <ul class="nav navbar-nav">
                <li class=""><a href="{ % url 'index' % }">Home</a></li>
                <li class=""><a href="{ % url 'UserLogin' % }">User</a></li>
                <li class=""><a href="{ % url 'AdminLogin' % }">Admin</a></li>
                <li class=""><a href="{ % url 'UserRegister' % }">Register</a></li>
              </ul>
            </div>
          </div>
        </div>
      </nav>
      <div class="container">
        <div class="row">
          <div class="banner-info">
            <div class="banner-logo text-center">
              </div>

```

Admin side admin_base.html:

```
<meta name="keywords" content="free website templates, free bootstrap themes, free
template, free bootstrap, free website template">
```

```
<link rel="stylesheet" type="text/css"
href="https://fonts.googleapis.com/css?family=Open+Sans|Raleway|Candal">
<link rel="stylesheet" type="text/css" href="{ % static 'css/font-awesome.min.css'
% } ">
<link rel="stylesheet" type="text/css" href="{ % static 'css/bootstrap.min.css' % } ">
<link rel="stylesheet" type="text/css" href="{ % static 'css/style.css' % } ">
</head>
```

```
<body id="myPage" data-spy="scroll" data-target=".navbar" data-offset="60">
<section id="banner" class="banner">
<div class="bg-color">
<nav class="navbar navbar-default navbar-fixed-top">
<div class="container">
<div class="col-md-12">
<div class="navbar-header">
<button type="button" class="navbar-toggle" data-toggle="collapse" data-
target="#myNavbar">
<span class="icon-bar"></span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
</button>
<h1 class="white">MACHINE LEARNING IN FAKE NEWS</h1>
</div>
<div class="collapse navbar-collapse navbar-right" id="myNavbar">
<ul class="nav navbar-nav">
<li class=""><a href="{ % url 'AdminHome' % }">Home</a></li>
<li class=""><a href="{ % url 'RegisterUsersView' % }">User
Details</a></li>
<li class=""><a href="{ % url 'adminResults' % }">Results</a></li>
<li class=""><a href="{ % url 'index' % }">Logout</a></li>
```

```

    </div>
  </div>
</footer>
<script src="{ % static 'js/jquery.min.js' % }"></script>
<script src="{ % static 'js/jquery.easing.min.js' % }"></script>
<script src="{ % static 'js/bootstrap.min.js' % }"></script>
<script src="{ % static 'js/custom.js' % }"></script>
<script src="{ % static 'contactform/contactform.js' % }"></script>
</body>
</html>

```

Admin side view_registered_users.html:

```

{% extends 'admins/adminbase.html' %}
{% block contents %}

<section id="intro">
  <div class="intro-text">
    <p>
      <h3 align="center" style="color:white;">View RegisterUser Details</h3>
    </p>
    <br>
    <table class="table table-bordered bg-light text-dark">
      <thead>
        <tr style="color:red;font-weight:bold;">
          <th>S.No</th>
          <th>Name</th>
          <th>Login ID</th>
          <th>Mobile</th>
          <th>Email</th>
          <th>Locality</th>
          <th>Status</th>
          <th>Activate</th>

```

```
</tr>
</thead>
<tbody>
  {% for i in data %}
  <tr style="color: white">
    <td>{{ forloop.counter }}</td>
    <td>{{ i.name }}</td>
    <td>{{ i.loginid }}</td>
    <td>{{ i.mobile }}</td>
    <td>{{ i.email }}</td>
    <td>{{ i.locality }}</td>
    <td>{{ i.status }}</td>

    {% if i.status == 'waiting' %}
    <td><a class="btn-link" href="/ActivaUsers/?uid={{ i.id }}"
style="color:DARKBLUE">Activate</a></td>
    {% else %}
    <td>Activated</td>
    {% endif %}
  </tr>
  {% endfor %}
</tbody>
</table>
</div>
</section>
{% endblock %}
```

Index.html:

```
{% extends 'base.html' %}

{% load static %}


{% block contents %}
<h3 style="color:white">Role of machine learning in fakenews</h3>
{% endblock %}
```