<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>Fake News📰 Detection System</title>

  <link href="https://fonts.googleapis.com/css?family=Open+Sans|Pacifico&display=swap" rel="stylesheet">

  <style>

    /\* General styles \*/

    \*,

    \*::before,

    \*::after {

      box-sizing: border-box;

    }

    body {

      margin: 0;

      padding: 0;

      font-family: 'Open Sans', sans-serif;

      color: #fff;

      font-size: 18px;

      text-align: center;

      letter-spacing: 1.2px;

      background: linear-gradient(135deg, #ff5f6d, #ffc371);

      display: flex;

      justify-content: center;

      align-items: center;

      min-height: 100vh;

    }

    .login {

      width: 400px;

      padding: 40px 20px; /\* Adjusted padding \*/

      background: rgba(0, 0, 0, 0.8);

      border-radius: 8px;

      box-shadow: 0 0 20px rgba(0, 0, 0, 0.4);

      animation: fadeIn 1s ease-in-out;

    }

    .login h1 {

      margin-bottom: 30px;

      font-family: 'Pacifico', cursive;

      font-size: 48px;

      text-shadow: 0 0 10px rgba(0, 0, 0, 0.5);

    }

    textarea {

      width: calc(100% - 40px);

      height: 150px;

      margin-bottom: 20px;

      padding: 10px;

      background: rgba(255, 255, 255, 0.1);

      border: none;

      outline: none;

      font-size: 18px;

      color: #fff;

      border-radius: 8px;

      resize: none;

      transition: background-color 0.3s ease;

    }

    textarea:focus {

      background: rgba(255, 255, 255, 0.2);

    }

    .btn {

      display: inline-block;

      padding: 12px 40px; /\* Adjusted padding \*/

      border: none;

      border-radius: 8px;

      background: #ff5f6d;

      color: #fff;

      font-size: 18px;

      text-align: center;

      text-decoration: none;

      cursor: pointer;

      transition: background-color 0.3s ease;

    }

    .btn:hover {

      background: #ff7b69;

    }

    .results {

      margin-top: 20px;

      opacity: 0;

      animation: fadeIn 1s ease-in-out forwards;

    }

    .results h2 {

      font-family: 'Pacifico', cursive;

      font-size: 32px;

      text-shadow: 0 0 10px rgba(0, 0, 0, 0.5);

      margin: 0;

    }

    .results p {

      font-size: 18px;

      margin-top: 10px;

    }

    .results h2.warning {

      color: red;

    }

    .results h2.success {

      color: #16112f;

    }

    /\* Keyframe animations \*/

    @keyframes fadeIn {

      from {

        opacity: 0;

      }

      to {

        opacity: 1;

      }

    }

  </style>

</head>

<body>

  <div class="login" id="loginContainer">

    <h1>Fake News📰 Detector</h1>

    <form action="{{ url\_for('predict')}}" method="POST">

      <textarea name="message" rows="6" cols="50" required="required" style="font-size: 18pt">{{ message }}</textarea>

      <br>

      <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>

    </form>

    <div class="results">

      {% if prediction == ['FAKE'] %}

      <h2 class="warning">Warning! 🚨</h2>

      <p>This news might be fake. Please verify before sharing.</p>

      <script>

        document.getElementById('loginContainer').style.background = '#ff5f6d';

      </script>

      {% elif prediction == ['REAL'] %}

      <h2 class="success">Success! ✅</h2>

      <p>This news appears to be authentic.</p>

      <script>

        document.getElementById('loginContainer').style.background = '#ffc371';

      </script>

      {% endif %}

    </div>

  </div>

</body>

</html>

from flask import Flask, render\_template, request

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.linear\_model import PassiveAggressiveClassifier

import pickle

import pandas as pd

from sklearn.model\_selection import train\_test\_split

app = Flask(\_\_name\_\_)

tfvect = TfidfVectorizer(stop\_words='english', max\_df=0.7)

loaded\_model = pickle.load(open('model.pkl', 'rb'))

dataframe = pd.read\_csv('news.csv')

x = dataframe['text']

y = dataframe['label']

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.2, random\_state=0)

def fake\_news\_det(news):

    tfid\_x\_train = tfvect.fit\_transform(x\_train)

    tfid\_x\_test = tfvect.transform(x\_test)

    input\_data = [news]

    vectorized\_input\_data = tfvect.transform(input\_data)

    prediction = loaded\_model.predict(vectorized\_input\_data)

    return prediction

@app.route('/')

def home():

    return render\_template('index.html', message="")

@app.route('/predict', methods=['POST'])

def predict():

    if request.method == 'POST':

        message = request.form['message']

        pred = fake\_news\_det(message)

        return render\_template('index.html', prediction=pred, message=message)

    else:

        return render\_template('index.html', prediction="Something went wrong", message="")

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True)