from flask import Flask, render\_template, request

import pandas as pd

import difflib

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.metrics.pairwise import cosine\_similarity

import requests

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

# Load dataset

movies\_data = pd.read\_csv('movies.csv')

selected\_features = ['genres', 'keywords', 'tagline', 'cast', 'director']

for feature in selected\_features:

    movies\_data[feature] = movies\_data[feature].fillna('')

movies\_data['combined\_features'] = (

    movies\_data['genres'] + ' ' +

    movies\_data['keywords'] + ' ' +

    movies\_data['tagline'] + ' ' +

    movies\_data['cast'] + ' ' +

    movies\_data['director']

)

vectorizer = TfidfVectorizer()

feature\_vectors = vectorizer.fit\_transform(movies\_data['combined\_features'])

similarity = cosine\_similarity(feature\_vectors)

# Fetch movie poster

def fetch\_movie\_poster(title):

    api\_key = '5e63a16c'  # Your OMDb API key

    url = f"http://www.omdbapi.com/?t={title}&apikey={api\_key}"

    response = requests.get(url)

    if response.status\_code == 200:

        data = response.json()

        poster = data.get('Poster', '')

        if poster and poster != "N/A":

            return poster

        else:

            print(f"No poster available for '{title}'")

    elif response.status\_code == 401:

        print("Unauthorized access to OMDb API. Check your API key or plan limits.")

    else:

        print(f"Error fetching poster for '{title}': {response.status\_code}")

    # Return a placeholder image

    return "https://via.placeholder.com/150"

# Route for homepage

@app.route('/')

def index():

    return render\_template('index.html')

# Route for recommendations

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

def recommend():

    movie\_name = request.form['movie\_name']

    list\_of\_all\_titles = movies\_data['title'].tolist()

    find\_close\_match = difflib.get\_close\_matches(movie\_name, list\_of\_all\_titles)

    if find\_close\_match:

        close\_match = find\_close\_match[0]

        index\_of\_the\_movie = movies\_data[movies\_data.title == close\_match].index[0]

        similarity\_score = list(enumerate(similarity[index\_of\_the\_movie]))

        sorted\_similar\_movies = sorted(similarity\_score, key=lambda x: x[1], reverse=True)

        recommendations = []

        similarity\_basis = {}  # Dictionary to track attribute similarity

        for movie in sorted\_similar\_movies[1:11]:  # Skip the first result (it's the input movie)

            index = movie[0]

            title\_from\_index = movies\_data.iloc[index]['title']

            genre = movies\_data.iloc[index]['genres']

            poster\_url = fetch\_movie\_poster(title\_from\_index)

            similarity\_score = round(movie[1] \* 100, 2)  # Convert similarity to percentage

            # Track which attributes match

            for feature in selected\_features:

                if movies\_data.iloc[index\_of\_the\_movie][feature] == movies\_data.iloc[index][feature]:

                    similarity\_basis[feature] = similarity\_basis.get(feature, 0) + 1

            recommendations.append({

                'title': title\_from\_index,

                'genre': genre,

                'poster': poster\_url,

                'similarity': similarity\_score

            })

        # Find the most similar attribute(s)

        most\_similar\_attributes = ", ".join(sorted(similarity\_basis, key=similarity\_basis.get, reverse=True)[:2])

        return render\_template('results.html', movie\_name=close\_match, recommendations=recommendations, basis=most\_similar\_attributes)

    return render\_template('results.html', movie\_name=movie\_name, recommendations=[], basis="unknown")

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

    app.run(debug=True)

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Movie Recommender</title>

    <link rel="stylesheet" href="{{ url\_for('static', filename='style.css') }}">

</head>

<body>

    <nav style="background-color: #333; padding: 10px; color: white; text-align: center;">

        <a href="/" style="color: white; text-decoration: none; font-size: 1.2em; margin-right: 20px;">Home</a>

        <span>Movie Recommender</span>

    </nav>

    <div style="text-align: center; margin-top: 50px;">

        <h1>Movie Recommendation System</h1>

        <form action="/recommend" method="post">

            <label for="movie\_name">Enter a movie name:</label><br>

            <input type="text" id="movie\_name" name="movie\_name" required><br><br>

            <button type="submit">Get Recommendations</button>

        </form>

    </div>

    <footer style="background-color: #f4f4f4; padding: 10px; text-align: center; margin-top: 50px; font-size: 0.9em;">

        &copy; 2025 Movie Recommender System | Built with Flask

    </footer>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Recommendations</title>

    <link rel="stylesheet" href="{{ url\_for('static', filename='style.css') }}">

</head>

<body>

    <nav style="background-color: #333; padding: 10px; color: white; text-align: center;">

        <a href="/" style="color: white; text-decoration: none; font-size: 1.2em; margin-right: 20px;">Home</a>

        <span>Movie Recommender</span>

    </nav>

    <div class="container">

        <h1>Recommendations for '{{ movie\_name }}'</h1>

        <h3>Based on similarity of: {{ basis }}</h3>

        {% if recommendations %}

            <div>

                {% for movie in recommendations %}

                    <div class="movie-card">

                        <img src="{{ movie.poster }}" alt="Poster for {{ movie.title }}">

                        <h3>{{ movie.title }}</h3>

                        <p><strong>Genre:</strong> {{ movie.genre }}</p>

                        <p><strong>Similarity:</strong> {{ movie.similarity }}%</p>

                    </div>

                {% endfor %}

            </div>

        {% else %}

            <p>No recommendations found for the given movie.</p>

        {% endif %}

        <br><a href="/">Go Back</a>

    </div>

    <footer style="background-color: #f4f4f4; padding: 10px; text-align: center; margin-top: 50px; font-size: 0.9em;">

        &copy; 2025 Movie Recommender System | Built with Flask

    </footer>

</body>

</html>

/\* General styles \*/

body {

    font-family: Arial, sans-serif;

    margin: 0;

    padding: 0;

    background-color: #f4f4f4;

    color: #333;

}

/\* Navigation bar \*/

nav {

    background-color: #333;

    color: white;

    padding: 10px 0;

    text-align: center;

    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);

}

nav a {

    color: white;

    text-decoration: none;

    font-size: 1.2em;

    margin: 0 15px;

}

nav a:hover {

    text-decoration: underline;

}

/\* Container styling \*/

.container {

    text-align: center;

    margin: 50px auto;

    max-width: 900px;

}

/\* Header styling \*/

h1 {

    color: #333;

    font-size: 2.5em;

    margin-bottom: 20px;

}

h3 {

    color: #555;

    font-size: 1.5em;

    margin-bottom: 20px;

}

/\* Form styling \*/

form {

    margin: 20px 0;

}

input[type="text"] {

    padding: 10px;

    width: 300px;

    font-size: 1em;

    border: 1px solid #ddd;

    border-radius: 5px;

    margin-bottom: 10px;

    box-shadow: inset 0 2px 4px rgba(0, 0, 0, 0.1);

}

button {

    padding: 12px 25px;

    font-size: 1em;

    color: white;

    background-color: #007bff;

    border: none;

    border-radius: 5px;

    cursor: pointer;

    transition: background-color 0.3s;

}

button:hover {

    background-color: #0056b3;

}

/\* Movie card styling \*/

.movie-card {

    display: inline-block;

    border: 1px solid #ddd;

    border-radius: 10px;

    margin: 15px;

    padding: 15px;

    background-color: white;

    width: 200px;

    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

    text-align: center;

    transition: transform 0.3s, box-shadow 0.3s;

}

.movie-card:hover {

    transform: scale(1.05);

    box-shadow: 0 8px 12px rgba(0, 0, 0, 0.2);

}

.movie-card img {

    width: 150px;

    height: auto;

    border-radius: 5px;

    margin-bottom: 10px;

}

.movie-card h3 {

    font-size: 1.2em;

    color: #333;

}

.movie-card p {

    margin: 5px 0;

    font-size: 0.9em;

    color: #666;

}

/\* Link styling \*/

a {

    text-decoration: none;

    color: #007bff;

    font-size: 1em;

    transition: color 0.3s;

}

a:hover {

    color: #0056b3;

}

/\* Footer styling \*/

footer {

    background-color: #f4f4f4;

    padding: 10px;

    text-align: center;

    margin-top: 50px;

    font-size: 0.9em;

    color: #999;

}