First time favorites feature

App.py

from flask import Flask, render\_template, request, session, jsonify

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\_\_)

app.secret\_key = "your\_secret\_key"  # Needed for Flask sessions

# 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

    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', 'GET'])

def recommend():

    if request.method == 'POST':

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

    else:

        movie\_name = request.args.get('movie\_name', '')

    session['last\_movie'] = movie\_name  # Save the last searched movie in the session

    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 = []

        for movie in sorted\_similar\_movies[1:11]:  # Skip the first result (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

            recommendations.append({

                'title': title\_from\_index,

                'genre': genre,

                'poster': poster\_url,

                'similarity': similarity\_score

            })

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

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

# Route to handle favorites

@app.route('/save\_favorite', methods=['POST'])

def save\_favorite():

    data = request.get\_json()

    movie\_title = data.get('movie\_title')

    remove = data.get('remove', False)

    favorites = session.get('favorites', [])

    if remove:

        if movie\_title in favorites:

            favorites.remove(movie\_title)

    else:

        if movie\_title not in favorites:

            favorites.append(movie\_title)

    session['favorites'] = favorites

    return jsonify({'success': True, 'favorites': favorites})

# Route to display favorites

@app.route('/favorites')

def favorites():

    favorites = session.get('favorites', [])

    favorite\_movies = []

    for title in favorites:

        movie = movies\_data[movies\_data['title'] == title].iloc[0]

        favorite\_movies.append({

            'title': movie['title'],

            'genre': movie['genres'],

            'poster': fetch\_movie\_poster(movie['title'])

        })

    return render\_template('favorites.html', favorite\_movies=favorite\_movies)

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

    app.run(debug=True)

Index.html

<!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>

Results.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') }}">

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css">

</head>

<body>

    <nav>

        <a href="/">Home</a>

        <a href="/favorites">Favorites</a>

    </nav>

    <div class="container">

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

        {% if recommendations %}

            <div class="movie-grid">

                {% for movie in recommendations %}

                    <div class="movie-card">

                        <div class="favorite-icon" data-title="{{ movie.title }}">

                            <i class="fa fa-heart-o"></i>

                        </div>

                        <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 %}

    </div>

    <script>

        document.addEventListener('DOMContentLoaded', () => {

            const favoriteIcons = document.querySelectorAll('.favorite-icon');

            favoriteIcons.forEach(icon => {

                icon.addEventListener('click', () => {

                    const title = icon.getAttribute('data-title');

                    const heartIcon = icon.querySelector('.fa');

                    // Toggle heart color

                    const isFavorite = heartIcon.classList.contains('fa-heart');

                    heartIcon.classList.toggle('fa-heart', !isFavorite);

                    heartIcon.classList.toggle('fa-heart-o', isFavorite);

                    // Send favorite state to the server

                    fetch('/save\_favorite', {

                        method: 'POST',

                        headers: {

                            'Content-Type': 'application/json',

                        },

                        body: JSON.stringify({

                            movie\_title: title,

                            remove: isFavorite

                        }),

                    });

                });

            });

        });

    </script>

</body>

</html>

Favorites.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Favorites</title>

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

</head>

<body>

    <nav>

        <a href="/">Home</a>

    </nav>

    <div class="container">

        <h1>Your Favorites</h1>

        {% if favorite\_movies %}

            <div class="movie-grid">

                {% for movie in favorite\_movies %}

                    <div class="movie-card">

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

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

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

                    </div>

                {% endfor %}

            </div>

        {% else %}

            <p>No favorites added yet.</p>

        {% endif %}

        <a href="/recommend?movie\_name={{ session.get('last\_movie') }}">Go Back</a>

    </div>

</body>

</html>

Style.css

/\* 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 \*/

.container {

    text-align: center;

    margin: 50px auto;

    max-width: 1200px;

}

/\* Headers \*/

h1 {

    color: #333;

    font-size: 2.5em;

    margin-bottom: 20px;

}

h3 {

    color: #555;

    font-size: 1.5em;

    margin-bottom: 20px;

}

/\* Form \*/

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 Grid \*/

.movie-grid {

    display: flex;

    flex-wrap: wrap;

    justify-content: center;

    gap: 20px;

}

/\* Movie Card \*/

.movie-card {

    position: relative;

    border: 1px solid #ddd;

    border-radius: 10px;

    padding: 15px;

    background-color: white;

    width: 220px;

    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: 100%;

    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;

}

/\* Favorites Button (Heart Icon) \*/

.favorite-icon {

    position: absolute;

    top: 10px;

    right: 10px;

    background-color: rgb(250, 249, 249);

    border: 1px solid #cfcfcf;

    border-radius: 50%;

    width: 28px;

    height: 28px;

    display: flex;

    align-items: center;

    justify-content: center;

    cursor: pointer;

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

    transition: transform 0.2s;

}

.favorite-icon:hover {

    transform: scale(1.1);

}

.favorite-icon .fa {

    font-size: 18px;

    color: gray;

}

.favorite-icon .fa-heart {

    color: red;

}

/\* Links \*/

a {

    text-decoration: none;

    color: #007bff;

    font-size: 1em;

    transition: color 0.3s;

}

a:hover {

    color: #0056b3;

}

/\* Footer \*/

footer {

    background-color: #f4f4f4;

    padding: 10px;

    text-align: center;

    margin-top: 50px;

    font-size: 0.9em;

    color: #999;

}