<SKIPPED\_HTML\_EDIT\_MODE></SKIPPED\_HTML\_EDIT\_MODE>

Here's the complete implementation:

```html

`movie\_recommender.html`

```html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>DevOps Movie Recommender</title>

<style>

:root {

--primary-color: #2c3e50;

--secondary-color: #3498db;

--accent-color: #e74c3c;

--light-color: #ecf0f1;

--dark-color: #2c3e50;

}

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

}

body {

background-color: #f5f5f5;

color: var(--dark-color);

line-height: 1.6;

}

.container {

max-width: 1200px;

margin: 0 auto;

padding: 20px;

}

header {

background-color: var(--primary-color);

color: white;

padding: 20px 0;

text-align: center;

margin-bottom: 30px;

border-radius: 0 0 10px 10px;

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

}

header h1 {

font-size: 2.5rem;

margin-bottom: 10px;

}

.tagline {

font-size: 1.2rem;

opacity: 0.8;

}

.search-section {

display: flex;

justify-content: center;

margin-bottom: 30px;

}

.search-container {

width: 100%;

max-width: 600px;

position: relative;

}

#movie-search {

width: 100%;

padding: 15px 20px;

border: 2px solid var(--secondary-color);

border-radius: 30px;

font-size: 1rem;

outline: none;

transition: all 0.3s ease;

}

#movie-search:focus {

box-shadow: 0 0 10px rgba(52, 152, 219, 0.5);

}

.search-btn {

position: absolute;

right: 5px;

top: 5px;

background-color: var(--secondary-color);

color: white;

border: none;

padding: 10px 20px;

border-radius: 25px;

cursor: pointer;

transition: all 0.3s ease;

}

.search-btn:hover {

background-color: #2980b9;

transform: scale(1.05);

}

.filter-section {

display: flex;

justify-content: center;

gap: 15px;

margin-bottom: 30px;

flex-wrap: wrap;

}

.filter-btn {

background-color: white;

border: 1px solid var(--secondary-color);

color: var(--secondary-color);

padding: 8px 15px;

border-radius: 20px;

cursor: pointer;

transition: all 0.3s ease;

}

.filter-btn.active {

background-color: var(--secondary-color);

color: white;

}

.filter-btn:hover {

background-color: var(--secondary-color);

color: white;

}

.movie-grid {

display: grid;

grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));

gap: 25px;

}

.movie-card {

background-color: white;

border-radius: 10px;

overflow: hidden;

box-shadow: 0 3px 10px rgba(0, 0, 0, 0.1);

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

}

.movie-card:hover {

transform: translateY(-10px);

box-shadow: 0 10px 20px rgba(0, 0, 0, 0.2);

}

.movie-poster {

width: 100%;

height: 350px;

object-fit: cover;

display: block;

}

.movie-info {

padding: 15px;

}

.movie-title {

font-size: 1.2rem;

margin-bottom: 10px;

color: var(--primary-color);

}

.movie-meta {

display: flex;

justify-content: space-between;

margin-bottom: 10px;

font-size: 0.9rem;

color: #7f8c8d;

}

.movie-rating {

color: #f39c12;

font-weight: bold;

}

.movie-overview {

font-size: 0.9rem;

margin-bottom: 15px;

display: -webkit-box;

-webkit-line-clamp: 3;

-webkit-box-orient: vertical;

overflow: hidden;

text-overflow: ellipsis;

}

.recommend-btn {

display: block;

width: 100%;

padding: 10px;

background-color: var(--secondary-color);

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

transition: background-color 0.3s ease;

}

.recommend-btn:hover {

background-color: #2980b9;

}

.loading {

text-align: center;

font-size: 1.5rem;

margin: 50px 0;

color: var(--secondary-color);

}

.error-message {

text-align: center;

color: var(--accent-color);

margin: 50px 0;

font-size: 1.2rem;

}

footer {

text-align: center;

padding: 20px;

margin-top: 50px;

font-size: 0.9rem;

color: #7f8c8d;

}

@media (max-width: 768px) {

header h1 {

font-size: 2rem;

}

.tagline {

font-size: 1rem;

}

.movie-grid {

grid-template-columns: repeat(auto-fill, minmax(200px, 1fr));

}

.movie-poster {

height: 300px;

}

}

@media (max-width: 480px) {

header h1 {

font-size: 1.5rem;

}

.movie-grid {

grid-template-columns: 1fr;

}

}

</style>

</head>

<body>

<header>

<div class="container">

<h1>DevOps Movie Recommender</h1>

<p class="tagline">Discover your next favorite movie powered by DevOps</p>

</div>

</header>

<main class="container">

<section class="search-section">

<div class="search-container">

<input type="text" id="movie-search" placeholder="Search for movies...">

<button class="search-btn">Search</button>

</div>

</section>

<section class="filter-section">

<button class="filter-btn active" data-genre="all">All</button>

<button class="filter-btn" data-genre="28">Action</button>

<button class="filter-btn" data-genre="12">Adventure</button>

<button class="filter-btn" data-genre="16">Animation</button>

<button class="filter-btn" data-genre="35">Comedy</button>

<button class="filter-btn" data-genre="18">Drama</button>

<button class="filter-btn" data-genre="878">Sci-Fi</button>

</section>

<section id="movie-results">

<!-- Movies will be loaded here -->

<div class="loading">Loading recommendations...</div>

</section>

</main>

<footer>

<div class="container">

<p>DevOps Movie Recommender &copy; 2023 | Powered by TMDB API</p>

</div>

</footer>

<script>

document.addEventListener('DOMContentLoaded', function() {

const apiKey = 'd931f7a2e87f71392fd1c5c6e3b5aacd'; // Public demo API key

const baseUrl = 'https://api.themoviedb.org/3';

const movieResults = document.getElementById('movie-results');

const searchInput = document.getElementById('movie-search');

const searchBtn = document.querySelector('.search-btn');

const filterBtns = document.querySelectorAll('.filter-btn');

let currentGenre = 'all';

// Load popular movies on page load

fetchPopularMovies();

// Event listeners

searchBtn.addEventListener('click', handleSearch);

searchInput.addEventListener('keypress', function(e) {

if (e.key === 'Enter') handleSearch();

});

filterBtns.forEach(btn => {

btn.addEventListener('click', function() {

filterBtns.forEach(b => b.classList.remove('active'));

this.classList.add('active');

currentGenre = this.dataset.genre;

if (searchInput.value.trim() === '') {

if (currentGenre === 'all') {

fetchPopularMovies();

} else {

fetchMoviesByGenre(currentGenre);

}

} else {

handleSearch();

}

});

});

function handleSearch() {

const query = searchInput.value.trim();

if (query === '') {

if (currentGenre === 'all') {

fetchPopularMovies();

} else {

fetchMoviesByGenre(currentGenre);

}

} else {

fetchMoviesBySearch(query);

}

}

async function fetchPopularMovies() {

try {

movieResults.innerHTML = '<div class="loading">Loading recommendations...</div>';

const response = await fetch(`${baseUrl}/movie/popular?api\_key=${apiKey}&language=en-US&page=1`);

const data = await response.json();

displayMovies(data.results);

} catch (error) {

movieResults.innerHTML = '<div class="error-message">Failed to load movies. Please try again later.</div>';

console.error('Error fetching popular movies:', error);

}

}

async function fetchMoviesByGenre(genreId) {

try {

movieResults.innerHTML = '<div class="loading">Loading movies...</div>';

let url;

if (genreId === 'all') {

url = `${baseUrl}/movie/popular?api\_key=${apiKey}&language=en-US&page=1`;

} else {

url = `${baseUrl}/discover/movie?api\_key=${apiKey}&language=en-US&sort\_by=popularity.desc&include\_adult=false&include\_video=false&page=1&with\_genres=${genreId}`;

}

const response = await fetch(url);

const data = await response.json();

displayMovies(data.results);

} catch (error) {

movieResults.innerHTML = '<div class="error-message">Failed to load movies. Please try again later.</div>';

console.error('Error fetching movies by genre:', error);

}

}

async function fetchMoviesBySearch(query) {

try {

movieResults.innerHTML = '<div class="loading">Searching movies...</div>';

let url = `${baseUrl}/search/movie?api\_key=${apiKey}&language=en-US&query=${encodeURIComponent(query)}&page=1&include\_adult=false`;

if (currentGenre !== 'all') {

url += `&with\_genres=${currentGenre}`;

}

const response = await fetch(url);

const data = await response.json();

if (data.results.length === 0) {

movieResults.innerHTML = '<div class="error-message">No movies found. Try a different search.</div>';

} else {

displayMovies(data.results);

}

} catch (error) {

movieResults.innerHTML = '<div class="error-message">Failed to search movies. Please try again later.</div>';

console.error('Error searching movies:', error);

}

}

function displayMovies(movies) {

if (!movies || movies.length === 0) {

movieResults.innerHTML = '<div class="error-message">No movies found. Try a different search.</div>';

return;

}

movieResults.innerHTML = '<div class="movie-grid"></div>';

const movieGrid = document.querySelector('.movie-grid');

movies.forEach(movie => {

if (!movie.poster\_path) return; // Skip movies without posters

const movieCard = document.createElement('div');

movieCard.className = 'movie-card';

const posterUrl = movie.poster\_path

? `https://image.tmdb.org/t/p/w500${movie.poster\_path}`

: 'https://placehold.co/500x750?text=No+Poster';

const releaseYear = movie.release\_date ? new Date(movie.release\_date).getFullYear() : '';

movieCard.innerHTML = `

<img src="${posterUrl}" alt="${movie.title} poster" class="movie-poster" onerror="this.src='https://placehold.co/500x750?text=Poster+Not+Available'">

<div class="movie-info">

<h3 class="movie-title">${movie.title}</h3>

<div class="movie-meta">

<span class="movie-year">${releaseYear}</span>

<span class="movie-rating">${movie.vote\_average.toFixed(1)}★</span>

</div>

<p class="movie-overview">${movie.overview || 'No description available.'}</p>

<button class="recommend-btn" data-id="${movie.id}">Get Similar</button>

</div>

`;

movieGrid.appendChild(movieCard);

});

// Add event listeners to recommendation buttons

document.querySelectorAll('.recommend-btn').forEach(btn => {

btn.addEventListener('click', function() {

fetchSimilarMovies(this.dataset.id);

});

});

}

async function fetchSimilarMovies(movieId) {

try {

movieResults.innerHTML = '<div class="loading">Finding similar movies...</div>';

const response = await fetch(`${baseUrl}/movie/${movieId}/similar?api\_key=${apiKey}&language=en-US&page=1`);

const data = await response.json();

if (data.results.length === 0) {

movieResults.innerHTML = '<div class="error-message">No similar movies found.</div>';

} else {

displayMovies(data.results);

}

} catch (error) {

movieResults.innerHTML = '<div class="error-message">Failed to find similar movies. Please try again later.</div>';

console.error('Error fetching similar movies:', error);

}

}

});

</script>

</body>

</html>

```

🎨 \*\*Generating images based on descriptions...\*\*

✅ \*\*Images generated successfully! Here's your updated content:\*\*

`movie\_recommender.html`

```html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>DevOps Movie Recommender</title>

<style>

:root {

--primary-color: #2c3e50;

--secondary-color: #3498db;

--accent-color: #e74c3c;

--light-color: #ecf0f1;

--dark-color: #2c3e50;

}

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

}

body {

background-color: #f5f5f5;

color: var(--dark-color);

line-height: 1.6;

}

.container {

max-width: 1200px;

margin: 0 auto;

padding: 20px;

}

header {

background-color: var(--primary-color);

color: white;

padding: 20px 0;

text-align: center;

margin-bottom: 30px;

border-radius: 0 0 10px 10px;

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

}

header h1 {

font-size: 2.5rem;

margin-bottom: 10px;

}

.tagline {

font-size: 1.2rem;

opacity: 0.8;

}

.search-section {

display: flex;

justify-content: center;

margin-bottom: 30px;

}

.search-container {

width: 100%;

max-width: 600px;

position: relative;

}

#movie-search {

width: 100%;

padding: 15px 20px;

border: 2px solid var(--secondary-color);

border-radius: 30px;

font-size: 1rem;

outline: none;

transition: all 0.3s ease;

}

#movie-search:focus {

box-shadow: 0 0 10px rgba(52, 152, 219, 0.5);

}

.search-btn {

position: absolute;

right: 5px;

top: 5px;

background-color: var(--secondary-color);

color: white;

border: none;

padding: 10px 20px;

border-radius: 25px;

cursor: pointer;

transition: all 0.3s ease;

}

.search-btn:hover {

background-color: #2980b9;

transform: scale(1.05);

}

.filter-section {

display: flex;

justify-content: center;

gap: 15px;

margin-bottom: 30px;

flex-wrap: wrap;

}

.filter-btn {

background-color: white;

border: 1px solid var(--secondary-color);

color: var(--secondary-color);

padding: 8px 15px;

border-radius: 20px;

cursor: pointer;

transition: all 0.3s ease;

}

.filter-btn.active {

background-color: var(--secondary-color);

color: white;

}

.filter-btn:hover {

background-color: var(--secondary-color);

color: white;

}

.movie-grid {

display: grid;

grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));

gap: 25px;

}

.movie-card {

background-color: white;

border-radius: 10px;

overflow: hidden;

box-shadow: 0 3px 10px rgba(0, 0, 0, 0.1);

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

}

.movie-card:hover {

transform: translateY(-10px);

box-shadow: 0 10px 20px rgba(0, 0, 0, 0.2);

}

.movie-poster {

width: 100%;

height: 350px;

object-fit: cover;

display: block;

}

.movie-info {

padding: 15px;

}

.movie-title {

font-size: 1.2rem;

margin-bottom: 10px;

color: var(--primary-color);

}

.movie-meta {

display: flex;

justify-content: space-between;

margin-bottom: 10px;

font-size: 0.9rem;

color: #7f8c8d;

}

.movie-rating {

color: #f39c12;

font-weight: bold;

}

.movie-overview {

font-size: 0.9rem;

margin-bottom: 15px;

display: -webkit-box;

-webkit-line-clamp: 3;

-webkit-box-orient: vertical;

overflow: hidden;

text-overflow: ellipsis;

}

.recommend-btn {

display: block;

width: 100%;

padding: 10px;

background-color: var(--secondary-color);

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

transition: background-color 0.3s ease;

}

.recommend-btn:hover {

background-color: #2980b9;

}

.loading {

text-align: center;

font-size: 1.5rem;

margin: 50px 0;

color: var(--secondary-color);

}

.error-message {

text-align: center;

color: var(--accent-color);

margin: 50px 0;

font-size: 1.2rem;

}

footer {

text-align: center;

padding: 20px;

margin-top: 50px;

font-size: 0.9rem;

color: #7f8c8d;

}

@media (max-width: 768px) {

header h1 {

font-size: 2rem;

}

.tagline {

font-size: 1rem;

}

.movie-grid {

grid-template-columns: repeat(auto-fill, minmax(200px, 1fr));

}

.movie-poster {

height: 300px;

}

}

@media (max-width: 480px) {

header h1 {

font-size: 1.5rem;

}

.movie-grid {

grid-template-columns: 1fr;

}

}

</style>

</head>

<body>

<header>

<div class="container">

<h1>DevOps Movie Recommender</h1>

<p class="tagline">Discover your next favorite movie powered by DevOps</p>

</div>

</header>

<main class="container">

<section class="search-section">

<div class="search-container">

<input type="text" id="movie-search" placeholder="Search for movies...">

<button class="search-btn">Search</button>

</div>

</section>

<section class="filter-section">

<button class="filter-btn active" data-genre="all">All</button>

<button class="filter-btn" data-genre="28">Action</button>

<button class="filter-btn" data-genre="12">Adventure</button>

<button class="filter-btn" data-genre="16">Animation</button>

<button class="filter-btn" data-genre="35">Comedy</button>

<button class="filter-btn" data-genre="18">Drama</button>

<button class="filter-btn" data-genre="878">Sci-Fi</button>

</section>

<section id="movie-results">

<!-- Movies will be loaded here -->

<div class="loading">Loading recommendations...</div>

</section>

</main>

<footer>

<div class="container">

<p>DevOps Movie Recommender © 2023 | Powered by TMDB API</p>

</div>

</footer>

<script>

document.addEventListener('DOMContentLoaded', function() {

const apiKey = 'd931f7a2e87f71392fd1c5c6e3b5aacd'; // Public demo API key

const baseUrl = 'https://api.themoviedb.org/3';

const movieResults = document.getElementById('movie-results');

const searchInput = document.getElementById('movie-search');

const searchBtn = document.querySelector('.search-btn');

const filterBtns = document.querySelectorAll('.filter-btn');

let currentGenre = 'all';

// Load popular movies on page load

fetchPopularMovies();

// Event listeners

searchBtn.addEventListener('click', handleSearch);

searchInput.addEventListener('keypress', function(e) {

if (e.key === 'Enter') handleSearch();

});

filterBtns.forEach(btn => {

btn.addEventListener('click', function() {

filterBtns.forEach(b => b.classList.remove('active'));

this.classList.add('active');

currentGenre = this.dataset.genre;

if (searchInput.value.trim() === '') {

if (currentGenre === 'all') {

fetchPopularMovies();

} else {

fetchMoviesByGenre(currentGenre);

}

} else {

handleSearch();

}

});

});

function handleSearch() {

const query = searchInput.value.trim();

if (query === '') {

if (currentGenre === 'all') {

fetchPopularMovies();

} else {

fetchMoviesByGenre(currentGenre);

}

} else {

fetchMoviesBySearch(query);

}

}

async function fetchPopularMovies() {

try {

movieResults.innerHTML = '<div class="loading">Loading recommendations...</div>';

const response = await fetch(`${baseUrl}/movie/popular?api\_key=${apiKey}&language=en-US&page=1`);

const data = await response.json();

displayMovies(data.results);

} catch (error) {

movieResults.innerHTML = '<div class="error-message">Failed to load movies. Please try again later.</div>';

console.error('Error fetching popular movies:', error);

}

}

async function fetchMoviesByGenre(genreId) {

try {

movieResults.innerHTML = '<div class="loading">Loading movies...</div>';

let url;

if (genreId === 'all') {

url = `${baseUrl}/movie/popular?api\_key=${apiKey}&language=en-US&page=1`;

} else {

url = `${baseUrl}/discover/movie?api\_key=${apiKey}&language=en-US&sort\_by=popularity.desc&include\_adult=false&include\_video=false&page=1&with\_genres=${genreId}`;

}

const response = await fetch(url);

const data = await response.json();

displayMovies(data.results);

} catch (error) {

movieResults.innerHTML = '<div class="error-message">Failed to load movies. Please try again later.</div>';

console.error('Error fetching movies by genre:', error);

}

}

async function fetchMoviesBySearch(query) {

try {

movieResults.innerHTML = '<div class="loading">Searching movies...</div>';

let url = `${baseUrl}/search/movie?api\_key=${apiKey}&language=en-US&query=${encodeURIComponent(query)}&page=1&include\_adult=false`;

if (currentGenre !== 'all') {

url += `&with\_genres=${currentGenre}`;

}

const response = await fetch(url);

const data = await response.json();

if (data.results.length === 0) {

movieResults.innerHTML = '<div class="error-message">No movies found. Try a different search.</div>';

} else {

displayMovies(data.results);

}

} catch (error) {

movieResults.innerHTML = '<div class="error-message">Failed to search movies. Please try again later.</div>';

console.error('Error searching movies:', error);

}

}

function displayMovies(movies) {

if (!movies || movies.length === 0) {

movieResults.innerHTML = '<div class="error-message">No movies found. Try a different search.</div>';

return;

}

movieResults.innerHTML = '<div class="movie-grid"></div>';

const movieGrid = document.querySelector('.movie-grid');

movies.forEach(movie => {

if (!movie.poster\_path) return; // Skip movies without posters

const movieCard = document.createElement('div');

movieCard.className = 'movie-card';

const posterUrl = movie.poster\_path

? `https://image.tmdb.org/t/p/w500${movie.poster\_path}`

: 'https://storage.googleapis.com/workspace-0f70711f-8b4e-4d94-86f1-2a93ccde5887/image/c34aff90-7fce-4bb7-9916-2ea09443c325.png';

const releaseYear = movie.release\_date ? new Date(movie.release\_date).getFullYear() : '';

movieCard.innerHTML = `

<img src="${posterUrl}" alt="${movie.title} poster" class="movie-poster" onerror="this.src='https://storage.googleapis.com/workspace-0f70711f-8b4e-4d94-86f1-2a93ccde5887/image/a9e0ddb0-3fe1-4f05-89c1-98f3425296bd.png'">

<div class="movie-info">

<h3 class="movie-title">${movie.title}</h3>

<div class="movie-meta">

<span class="movie-year">${releaseYear}</span>

<span class="movie-rating">${movie.vote\_average.toFixed(1)}★</span>

</div>

<p class="movie-overview">${movie.overview || 'No description available.'}</p>

<button class="recommend-btn" data-id="${movie.id}">Get Similar</button>

</div>

`;

movieGrid.appendChild(movieCard);

});

// Add event listeners to recommendation buttons

document.querySelectorAll('.recommend-btn').forEach(btn => {

btn.addEventListener('click', function() {

fetchSimilarMovies(this.dataset.id);

});

});

}

async function fetchSimilarMovies(movieId) {

try {

movieResults.innerHTML = '<div class="loading">Finding similar movies...</div>';

const response = await fetch(`${baseUrl}/movie/${movieId}/similar?api\_key=${apiKey}&language=en-US&page=1`);

const data = await response.json();

if (data.results.length === 0) {

movieResults.innerHTML = '<div class="error-message">No similar movies found.</div>';

} else {

displayMovies(data.results);

}

} catch (error) {

movieResults.innerHTML = '<div class="error-message">Failed to find similar movies. Please try again later.</div>';

console.error('Error fetching similar movies:', error);

}

}

});

</script>

</body>

</html>

```--- End of CSV data ---