index.ejs

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

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

<title>IMDb</title>

<link rel="stylesheet" href="/styles.css" />

</head>

<body>

<nav class="navbar">

<h1 class="logo">IMDb</h1>

<form action="/search" method="get" class="search-bar">

<input

type="text"

name="q"

placeholder="Search movies"

class="search-input"

/>

<button type="submit" class="search-button">Search</button>

</form>

</nav>

<main>

<section class="movie-list">

<h2 class="section-title">Movies</h2>

<div class="movies-grid">

<% movies.forEach(movie => { %>

<div class="movie-card">

<% if (movie.image) { %>

<img

src="<%= movie.image %>"

alt="<%= movie.name %>"

class="movie-poster"

/>

<% } else { %>

<div class="movie-placeholder">No Image</div>

<% } %>

<div class="movie-details">

<h3 class="movie-title"><%= movie.name %></h3>

<p class="movie-year">Year: <%= movie.year %></p>

<p class="movie-director">

Director: <%= movie.director %>

</p>

</div>

<a href="/movie/<%= movie.\_id %>" class="details-link"

>View Details</a

>

</div>

<% }) %>

</div>

</section>

<section class="add-movie-form">

<h2 class="section-title">Add a Movie</h2>

<form

action="/add-movie"

method="post"

enctype="multipart/form-data"

class="form-container"

>

<input

type="text"

name="name"

placeholder="Movie Name"

required

class="form-input"

/>

<input

type="text"

name="director"

placeholder="Director"

required

class="form-input"

/>

<input

type="text"

name="year"

placeholder="Year"

required

class="form-input"

/>

<input

type="text"

name="cast"

placeholder="Cast (comma-separated)"

required

class="form-input"

/>

<input

type="file"

name="image"

accept="image/\*"

class="form-input"

/>

<button type="submit" class="form-button">Add Movie</button>

</form>

</section>

</main>

</body>

</html>

movie.ejs

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

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

<title><%= movie.name %> - IMDb </title>

<link rel="stylesheet" href="/styles.css" />

</head>

<body>

<nav class="navbar">

<a href="/" class="back-link">Back to Movies</a>

</nav>

<main class="movie-details-page">

<div class="movie-details-container">

<% if (movie.image) { %>

<img

src="<%= movie.image %>"

alt="<%= movie.name %>"

class="movie-details-poster"

/>

<% } else { %>

<div class="movie-details-placeholder">No Image</div>

<% } %>

<div class="movie-info">

<h1 class="movie-details-title"><%= movie.name %></h1>

<p class="movie-details-year">Year: <%= movie.year %></p>

<p class="movie-details-director">

Director: <%= movie.director %>

</p>

<p class="movie-details-cast">

Cast: <%= movie.cast.join(', ') %>

</p>

</div>

</div>

<section class="comments-section">

<h2>Reviews</h2>

<% if (movie.comments && movie.comments.length > 0) { %>

<ul class="comments-list">

<% movie.comments.forEach(comment => { %>

<li class="comment-item"><%= comment %></li>

<% }) %>

</ul>

<% } else { %>

<p>No reviews yet. Be the first to write a review!</p>

<% } %>

<form

action="/add-comment/<%= movie.\_id %>"

method="post"

class="form-container"

>

<textarea

name="comment"

placeholder="Write your review..."

required

class="form-textarea"

></textarea>

<button type="submit" class="form-button">

Post Reviews

</button>

</form>

</section>

</main>

</body>

</html>

styles.css

body {

font-family: Arial, sans-serif;

background-color: #f5f5f5;

margin: 0;

padding: 0;

color: #333;

}

.navbar {

background-color: #ffcc00;

padding: 1rem;

display: flex;

justify-content: space-between;

align-items: center;

}

.logo {

font-size: 1.5rem;

font-weight: bold;

color: #000;

}

.search-bar {

display: flex;

gap: 0.5rem;

}

.search-input {

padding: 0.5rem;

font-size: 1rem;

}

.search-button {

background-color: #000;

color: #fff;

padding: 0.5rem 1rem;

border: none;

cursor: pointer;

}

.section-title {

text-align: center;

color: #333;

margin-top: 1rem;

font-size: 1.8rem;

}

.movie-list {

padding: 2rem;

}

.movies-grid {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(180px, 1fr));

gap: 1rem;

}

.movie-card {

background-color: #fff;

border-radius: 8px;

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

padding: 0.8rem;

text-align: center;

transition: transform 0.2s ease-in-out;

}

.movie-card:hover {

transform: scale(1.05);

}

.movie-poster {

width: 150px; /\* Set the desired width \*/

height: 225px; /\* Set the desired height to match a typical poster aspect ratio \*/

border-radius: 8px;

margin-bottom: 0.5rem;

object-fit: contain; /\* Ensures the whole image is displayed \*/

background-color: #f0f0f0; /\* Optional: Add a background color to handle empty spaces \*/

}

.movie-title {

font-size: 1.2rem;

color: #000;

margin-bottom: 0.3rem;

}

.movie-year,

.movie-director {

font-size: 0.9rem;

color: #555;

margin: 0.2rem 0;

}

.details-link {

display: inline-block;

margin-top: 0.5rem;

padding: 0.4rem 0.8rem;

background-color: #ffcc00;

color: #000;

text-decoration: none;

border-radius: 4px;

font-size: 0.9rem;

}

.details-link:hover {

background-color: #ffa500;

}

.add-movie-form,

.comments-section {

padding: 1.5rem;

background-color: #fff;

border-radius: 8px;

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

margin: 1rem auto;

max-width: 600px;

}

.form-container {

display: flex;

flex-direction: column;

gap: 0.8rem;

}

.form-input,

.form-textarea {

padding: 0.5rem;

font-size: 1rem;

border: 1px solid #ccc;

border-radius: 4px;

}

.form-button {

padding: 0.5rem 1rem;

background-color: #000;

color: #fff;

border: none;

cursor: pointer;

border-radius: 4px;

}

.form-button:hover {

background-color: #333;

}

.comments-list {

list-style-type: none;

padding: 0;

}

.comment-item {

padding: 0.5rem;

margin-bottom: 0.5rem;

background-color: #f9f9f9;

border-radius: 4px;

}

server.js

var express = require('express');

var app = express();

var mdb = require('mongodb').MongoClient;

var ObjectId = require('mongodb').ObjectId; // Import ObjectId

var bodyParser = require('body-parser');

var multer = require('multer'); // For file uploads

var path = require('path'); // To handle file paths

var con = new mdb("mongodb+srv://245122749304:Bhuvan$2005@cluster0.qzeb4.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0");

app.use(bodyParser.urlencoded({ extended: true }));

app.set('view engine', 'ejs');

app.set('views', 'views');

app.use(express.static('public')); // Serve static files

app.listen(1337);

// Configure multer for image uploads

var storage = multer.diskStorage({

destination: (req, file, cb) => {

cb(null, 'public/uploads/'); // Save images in public/uploads

},

filename: (req, file, cb) => {

cb(null, Date.now() + path.extname(file.originalname)); // Add a timestamp to the filename

}

});

var upload = multer({ storage: storage });

app.get('/', main); // Route to display movies

app.get('/movie/:id', movieDetails); // Route to display movie details

app.post('/add-comment/:id', addComment); // Route to add a comment

app.post('/add-movie', upload.single('image'), addMovie); // Route to add a new movie

app.get('/search', searchMovie); // Route to search movies

function main(req, res) {

con.db('imdb-clone').collection('movies').find().toArray()

.then(result => res.render('index', { movies: result }));

}

function movieDetails(req, res) {

let movieId = new ObjectId(req.params.id); // Use ObjectId here

con.db('imdb-clone').collection('movies').findOne({ \_id: movieId })

.then(movie => res.render('movie', { movie }));

}

function addComment(req, res) {

let movieId = new ObjectId(req.params.id); // Use ObjectId here

con.db('imdb-clone').collection('movies').updateOne(

{ \_id: movieId },

{ $push: { comments: req.body.comment } }

).then(() => res.redirect('/movie/' + req.params.id));

}

function addMovie(req, res) {

let newMovie = {

name: req.body.name,

director: req.body.director,

year: req.body.year,

cast: req.body.cast.split(',').map(name => name.trim()), // Split cast into an array

image: req.file ? '/uploads/' + req.file.filename : null, // Save image path

comments: []

};

con.db('imdb-clone').collection('movies').insertOne(newMovie)

.then(() => res.redirect('/'));

}

function searchMovie(req, res) {

let query = req.query.q || '';

con.db('imdb-clone').collection('movies').find({ name: new RegExp(query, 'i') }).toArray()

.then(result => res.render('index', { movies: result }));

}