MoviesApp Technical Documentation

Product Name:	MoviesApp
Product Version:	1.0.0
Owner:	Jesus Contreras
Product Phase:	Deployment
Current Date:	September 27, 2021

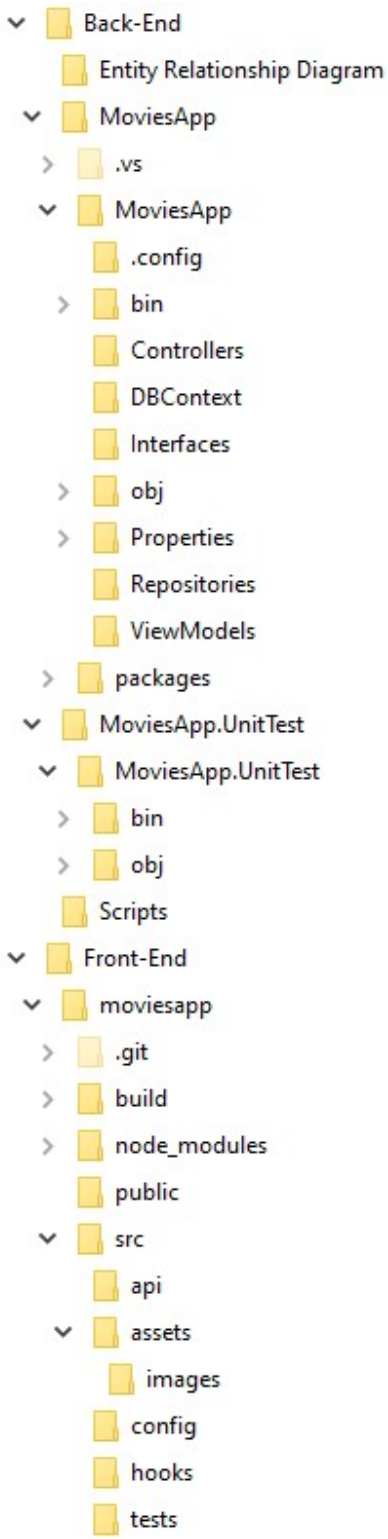
Product Overview

MoviesApp is a web-based application to browser movies information such as Title or Name, Director's Name, Production Year, Name of Actors (cast), Genres and the picture of their official posters.

Table of Contents

Project Folders and Files	2
Requirements Technical Sheet	3
Development Technical Sheet	3
Deployment Instructions: Back-End	4
Deployment Instructions: Front-End	5
Appendix	6

Project Folders and Files



Project is configured in Front-End and Back-End files.

Back-End folder contains Entity Relationship Diagram, Database scripts and source code for REST APIs (MoviesApp) and Tests (MoviesApp.UnitTest) C# projects.

Front-End folder contains source code for React functional components, custom hooks, helpers, HTML, CSS and images.

Requirements Technical Sheet

PC computer with Microsoft Windows 10 or Microsoft Windows Server as Web API server

PC computer with Linux, Microsoft Windows 10 or Microsoft Windows Server as Front-End Web server

Microsoft Internet Information Server as Web API server

Node JS and NPM HTTP Server as Front-End server

Microsoft SQL Server 2012 or above as Database Management System

Development Technical Sheet

Front-End

- MoviesApp Front-End layer was built upon the following technologies:
- React JS 17
- HTML 5
- CSS 3
- Bootstrap 4.5
- Animate.css 4.1
- Microsoft Visual Studio Code 1.6 as IDE

Back-End

- MoviesApp Back-End layer was built upon the following technologies:
- RESTful Web APIs made with C#
- Microsoft Internet Information Server as web server
- Microsoft Windows Server as OS server
- Microsoft SQL Server 2017 as Database Management System
- DDL, DML and Store Procedure scripts made with Microsoft Transact SQL
- Microsoft Visual Studio 2022 Preview 4 as IDE

Deployment Instructions

Back-End

Database setup

Create database using Microsoft SQL Server Management Studio IDE or use provided script “1 create_database.sql” located in folder \MoviesApp\Back-End\Scripts. If so, consider changing the following `FILENAME` parameters accordingly:

```
FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL14.SQLEXPRESS\MSSQL\DATA\MovieAppDatabase.mdf'
```

```
FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL14.SQLEXPRESS\MSSQL\DATA\MovieAppDatabase_log.ldf'
```

Recommended database name is “MoviesAppDatabase”.

Database data

Proceed to run provided scripts in folder \MoviesApp\Back-End\Scripts in this order:

2 create_tables.sql

3 insert_data.sql

4 create_sp_movie_getfbyfilter.sql

Web API site setup

To create web site in Windows IIS follow steps mentioned on this link [How to Deploy ASP.NET Core to IIS & How ASP.NET Core Hosting Works – Stackify]

Proceed to deploy MoviesApp Back-end files from folder \MoviesApp\Back-End\MoviesApp\MoviesApp\obj\publish to the created site. Change connection string in `appsettings.json` file according to SQL Server database connection parameters:

```
"MainConnection": "Server<your server name>; Database=MovieAppDatabase; User
id=<user>; Password=<password>;"
```

Front-End

Web server setup

Install Node JS following instructions in Node.js (nodejs.org)

Option 1: Install HTTP Server following instructions in http-server - npm (npmjs.com)

Proceed to create a folder and copy MoviesApp Front-End files provided in \MoviesApp\Front-End\moviesapp\build folder.

From that folder run the following command:

```
http-server . -p <your port>
```

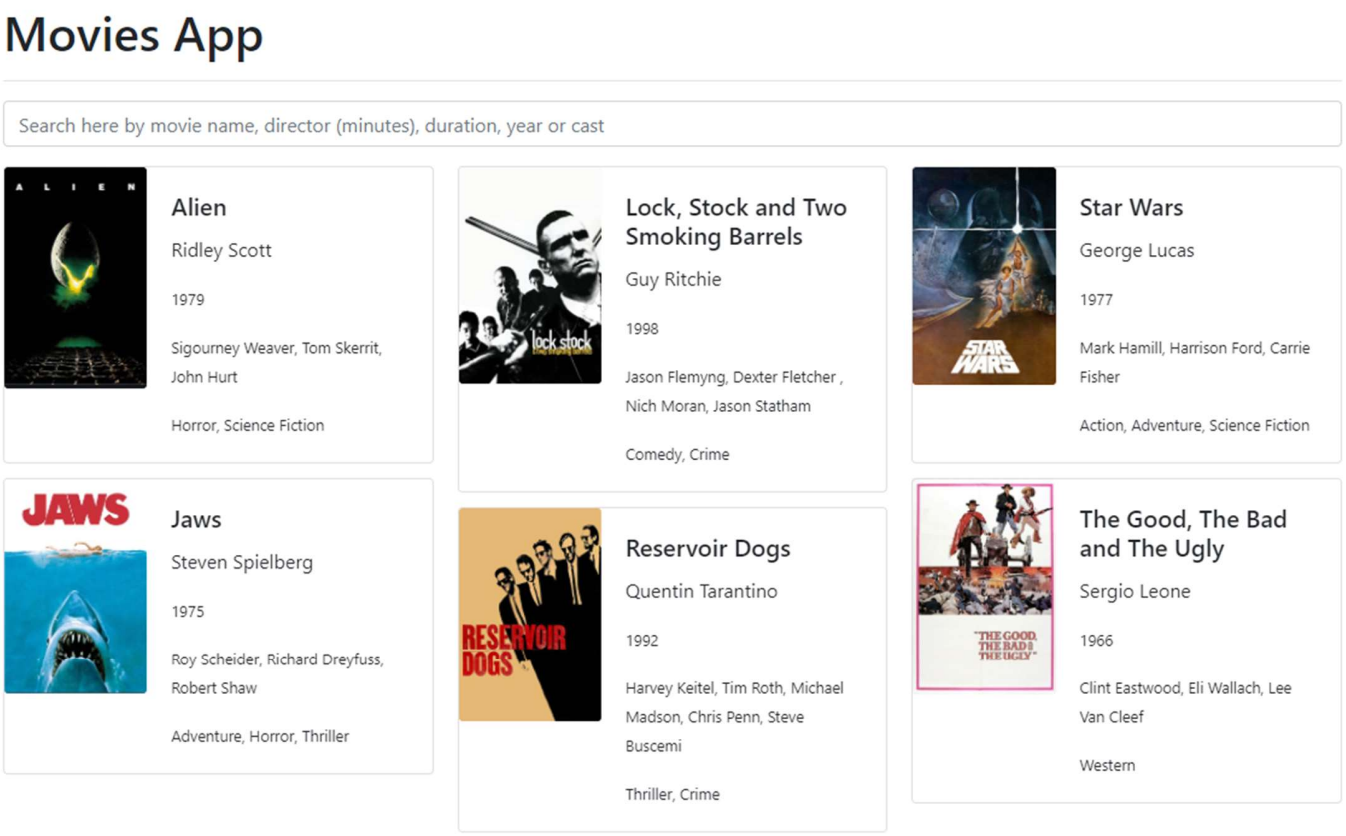
Option 2: Run following NPM commands:

```
npm install -g serve
```

```
serve -s build
```

APPENDIX

MoviesApp main screen



Entity Relationship Diagram