Movie Tracker Application

Getting Started (Source):

Website:

Website is hosted on Azure. The source code was developed in Visual Studio 2015, using MVC 5.

Additional source code is provided, that can be opened in Visual Studio. Please take note, the following Nuget Packages are required:

- Newtonsoft.Json
- PagedList.MVC

SQL Server Database:

The database DWSMovieTracker is hosted and run in Azure, but for local purposes, a database has been exported and provided. Along with script files of all stored procedures. A comprehensive list of stored procedures and functions will be listed at the end of this document. If you want to view the Azure database, please just send me your IP, and I will add it to the exception Firewall.

DatabaseName: DWSMovieTracker

Username: mtadmin

Password: ZF@2-L6pXvpbPca

Getting Started (Application):

The application can be viewed here:

http://dwsmovietracker.azurewebsites.net/

Additional Resources:

- Bootstrap theme: https://bootswatch.com/flatly/
- Website images sourced from: https://elements.envato.com/cinema-movie-set-4WP2W9
- Open Movie Database: http://www.omdbapi.com/

Website details:

The movie tracker website allows a user to perform CRUD functions and track the users movie collection.

Upon arriving, the user can either view a paged list of his existing movies, or quickly add a new one.

Adding a new movie:

A user can add his/her new movie easily, by entering the Title of a movie, and clicking the Get details button provided. This button, sends the Title to a Restful webservice hosted by the Open Source Database. And collects a Json string with movie details, and pushes it back to the frontend. This information can be changed before submitting to the database. Submitting a a new movie, calls the Upsert stored procedure, because we don't want duplicates, and the upsert stored procedure, when finding an entry in the Movie database, rather updates the field, than add a duplicate.

Editing/deleting a movie:

An existing movie can be edited by selecting the edit button, all details can be edited by means of the Upsert stored procedure. The delete function also makes use of the DeleteMovie stored procedure. Both of these actions deviate away from the standard MVC methods, because we want to make sure we delete/edit the relevant children data of the Movie table.

Searching Movies:

A global search field and button can be found in the layout page of the project. Allowing for any detail of the movie to be searched. The search function calls the Movie_Reader stored procedure, that returns all data in the database, linked to the search string.

SQL Scripts:

ClearDB:

Standard script that deletes all data in the database tables. For testing.

CreateDB:

Script displaying the creation of tables, and their primary keys in the database. This script is only to show the structures of tables.

Upsert_Movie

This stored procedure inserts a new movie file into the database, along with actors and genres associated to the movie. If the stored procedure finds an existing movie, the existing record will rather be updated. This stored procedure makes use of the Upsert_Actors and Upsert_Genres stored procedures.

Upsert_actors

This stored procedure allows the insert of actors linked to a certain movie. It receives the actors in a string, and uses the SplitString function to add actors. If an actor doesn't exist in the Actor table, it gets added, else it will get updated, and linked to the movie.

Upsert_Genres

This stored procedure works very much the same as the Upsert_Actors stored procedure, but this time only updates or adds Genres.

SplitString

This function is a generic function that receives a string, and character key, and splits the string into a table format according to the character key.

Delete_Movie

Stored procedure deletes a movie from the Movie table, along with the actors and genres linked to it.

Movie_Reader

This stored procedure accepts an input string, and checks both movie, actor and genre tables for anything related to it.

Original Requirements:

Overview

Build a web or desktop application in any Microsoft technology that allows me to keep track of all the movies in my collection. I have movies spread over DVDs, downloaded files etc. I want to be able to capture all the details of each my movies as well as be able to search them by title, year, actor, etc. as well track their location.

I want to be able to search my collection by director, actor etc and to view all the details of a title. I want to be able to edit existing titles, actors etc as well as add new ones and remove old ones. There are about 2000 titles in my collection.

Minimum deliverables

- Source code
- Deployed application or Instructions on how to get the app running.
- No CMS or CMS plugins may be used (Wordpress or Drupal)
- Any frameworks in the language of your choice may be used.
- A relational database must be used.
- While this is a simple interview exercise, please treat as you would any other project you would be delivering for DWS.