//Program.cs

using System;

using System.Collections.Generic;

namespace MovieScreeningApp

{

public class Program

{

public static void Main()

{

Menu();

}

public static void Menu()

{

Movie movie = new Movie();

MovieScreening movieOperations = new MovieScreening();

string loopInput = string.Empty;

do

{

Console.Write("Menu\n\n1. Add Movie" +

"\n2. Display Movies By Highest Viewers Rating" +

"\n3. Search Movies By Title's First Letter\n\nEnter your choice (1-3) : ");

int choice = int.Parse(Console.ReadLine());

switch (choice)

{

case 1:

Console.WriteLine();

Console.Write("Add Movie Details\n\n");

Console.Write("Title : ");

movie.Title = Console.ReadLine();

Console.Write("Duration(In Hrs) : ");

movie.Duration = float.Parse(Console.ReadLine());

Console.Write("Rating(1-5) : ");

movie.Rating = int.Parse(Console.ReadLine());

//Fill your code here

bool add = movieOperations.AddMovie(movie);

if(add)

Console.WriteLine("Movie Details Successfully Added to the Database");

else

Console.WriteLine("Failed to Add Movie Details");

break;

case 2:

Console.WriteLine("\nDisplay Movies By Highest Viewers' Rating\n");

//Fill your code here

IList<Movie> high = movieOperations.DisplayMoviesByHigestRating();

if(high==null || high.Count == 0)

Console.WriteLine("No Records Found");

else

{

Console.WriteLine("{0,-20}{1,-20}{2,-30}{3}", "Id", "Title", "Duration(In Hrs)", "Rating");

foreach (Movie m in high)

{

Console.WriteLine("{0,-20}{1,-20}{2,-30}{3}",m.Id, m.Title, m.Duration, m.Rating);

}

}

break;

case 3:

Console.Write("\nSearch Movies By First Letter : ");

string firstLetter = Console.ReadLine();

//Fill your code here

IList<Movie> search = movieOperations.SearchMoviesByStartLetter(firstLetter);

if(search == null || search.Count==0)

Console.WriteLine("No Movie(s) Found With The First Letter '{0}'",firstLetter);

else

{

Console.WriteLine("{0,-20}{1,-20}{2,-30}{3}", "Id", "Title", "Duration(In Hrs)", "Rating");

foreach (Movie m in search)

{

Console.WriteLine("{0,-20}{1,-20}{2,-30}{3}", m.Id, m.Title, m.Duration, m.Rating);

}

}

break;

default:

Console.WriteLine("Invalid Entry");

break;

}

Console.Write("\nDo You Want To Continue (Yes|No)? : ");

loopInput = Console.ReadLine();

}

while (loopInput.Equals("yes", StringComparison.InvariantCultureIgnoreCase));

}

}

}

//Movie.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace MovieScreeningApp

{

public class Movie

{

public int Id { get; set; }

public string Title { get; set; }

public float Duration { get; set; }

public int Rating { get; set; }

public Movie() { }

}

}

//MovieScreening,cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Configuration;

using System.Data;

using System.Data.SqlClient;

namespace MovieScreeningApp

{

public class MovieScreening

{

public MovieScreening() {

}

public string ConnectionString {

get {

return ConfigurationManager.ConnectionStrings["SqlCon"].ConnectionString;

}

}

public SqlConnection \_sqlConn { get; set; }

public bool AddMovie(Movie movie)

{

try

{

\_sqlConn = new SqlConnection(ConnectionString);

using (\_sqlConn)

{

string query = "insert into tblMovie (Name, Duration, Rating) values('" + movie.Title + "'," + movie.Duration + "," + movie.Rating + ")";

SqlCommand cmd = new SqlCommand(query, \_sqlConn);

\_sqlConn.Open();

int rowaffencted = cmd.ExecuteNonQuery();

\_sqlConn.Close();

if (rowaffencted > 0)

return true;

else

return false;

}

}catch(Exception)

{

return false;

}

}

public IList<Movie> DisplayMoviesByHigestRating()

{

\_sqlConn = new SqlConnection(ConnectionString);

using (\_sqlConn)

{

try

{

string query = "select \* from tblMovie where rating=(select Max(rating) from tblMovie)";

\_sqlConn.Open();

SqlCommand cmd = new SqlCommand(query, \_sqlConn);

SqlDataReader sdr = cmd.ExecuteReader();

List<Movie> highMovie = new List<Movie>();

while (sdr.Read())

{

Movie m = new Movie();

m.Id = (int)sdr.GetValue(0);

m.Title = sdr.GetValue(1).ToString();

m.Duration = (float)Convert.ToDouble(sdr.GetValue(2));

m.Rating = (int)sdr.GetValue(3);

highMovie.Add(m);

}

sdr.Close();

return highMovie;

}catch(Exception)

{

return null;

}

}

}

public IList<Movie> SearchMoviesByStartLetter(string searchedFirstLetter)

{

try

{

\_sqlConn = new SqlConnection(ConnectionString);

using (\_sqlConn)

{

string query = "select \* from tblMovie where name like ('" + searchedFirstLetter + "%')";

SqlCommand cmd = new SqlCommand(query, \_sqlConn);

\_sqlConn.Open();

SqlDataReader sdr = cmd.ExecuteReader();

List<Movie> searchMovie = new List<Movie>();

while (sdr.Read())

{

Movie m = new Movie();

m.Id = (int)sdr.GetValue(0);

m.Title = sdr.GetValue(1).ToString();

m.Duration = (float)Convert.ToDouble(sdr.GetValue(2));

m.Rating = (int)sdr.GetValue(3);

searchMovie.Add(m);

}

sdr.Close();

return searchMovie;

}

}

catch(Exception)

{

return new List<Movie>();

}

}

}

}