

## Web Fix Movie Library

MovieContext:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data.Entity;
using System.ComponentModel.DataAnnotations.Schema;

namespace WebFlixMovieLibrary
{
    //Do not change Connection Name under this class
    public class MovieContext:DbContext
    {
        public MovieContext():base("Name=DataConnection")
        {

        }

        //Fill your code here to create DbSet MovieGenres and Movies
        public virtual DbSet<MovieGenre> MovieGenres {get; set;}
        public virtual DbSet<MovieInformation> Movies {get; set;}

        protected override void OnModelCreating(DbModelBuilder modelBuilder)
        {
            //Fill your code here to implement Fluent API
            modelBuilder.Entity<MovieGenre>().ToTable("tblMovieGenre");
            modelBuilder.Entity<MovieInformation>().ToTable("tblMovie");

            modelBuilder.Entity<MovieGenre>().HasKey(g => g.MovieGenreId);
            modelBuilder.Entity<MovieGenre>().Property(g =>
g.MovieGenreId).HasDatabaseGeneratedOption(DatabaseGeneratedOption.None);
            modelBuilder.Entity<MovieInformation>().HasKey(m => m.MovieId);
            modelBuilder.Entity<MovieInformation>().Property(m =>
m.MovieId).HasDatabaseGeneratedOption(DatabaseGeneratedOption.None);

            modelBuilder.Entity<MovieInformation>()
                .HasRequired<MovieGenre>(m => m.Genre)
                .WithMany(g => g.MovieCollection)
                .HasForeignKey<int>(m => m.MovieCategoryId);

        }
    }
}
```

MovieFeedback:

```
using System;
namespace WebFlixMovieLibrary
{
    //Write your code
    public class MovieFeedback{
        public string GetFeedbackByRating(int rating){
            if(rating<2)
                return "Bad";
        }
    }
}
```

```

        else if (rating >= 2 && rating < 4)
            return "Average";

        else if (rating >= 4)
            return "Good";

        else
            return " ";
    }
}

```

MovieGenre:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace WebFlixMovieLibrary
{
    public class MovieGenre
    {
        //Fill your code here
        public int MovieGenreId {get; set;}
        public string MovieCategory {get; set;}
        public virtual ICollection<MovieInformation> MovieCollection {get; set;}
    }
}

```

MovieInformation:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace WebFlixMovieLibrary
{
    public class MovieInformation
    {
        //Fill your code here
        public int MovieId {get; set;}
        public int MovieCategoryId {get; set;}

        public virtual MovieGenre Genre {get; set;}

        public string MovieName {get; set;}
        public int ReleaseYear {get; set;}
        public string AudienceFeedback {get; set;}
    }
}

```

MovieRepository:

```

using System;
using System.Collections.Generic;
using System.Linq;

```

```

using System.Text;
using System.Threading.Tasks;

namespace WebFlixMovieLibrary
{
    public class MovieRepository
    {
        //Fill your code here
        public MovieGenre AddNewGenre(MovieGenre category){
            using(MovieContext cxt = new MovieContext()){
                cxt.MovieGenres.Add(category);
                cxt.SaveChanges();
                return category;
            }
        }

        public IList<MovieGenre> DisplayMovieGenres(){
            using(MovieContext cxt = new MovieContext()){
                return cxt.MovieGenres.ToList();
            }
        }

        public MovieInformation AddNewMovie(MovieInformation movie){
            using(MovieContext cxt = new MovieContext()){
                cxt.Movies.Add(movie);
                cxt.SaveChanges();
                return movie;
            }
        }

        public IList<MovieInformation> DisplayMovieLibrary(){
            using(MovieContext cxt = new MovieContext()){
                return cxt.Movies.ToList();
            }
        }
    }
}

```

Program:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace WebFlixMovieLibrary
{
    class Program
    {
        static void Main(string[] args)
        {
            MovieGenre movieGenre;
            MovieInformation movieInformation;
            MovieRepository movieRepository;
            string loopInput = string.Empty;
            int menuChoice = 0;
            do
            {
                Console.WriteLine("Welcome Admin to WebFlix Movie Library");
            }
        }
    }
}

```

```

Genre\n" +
        Console.WriteLine("Menu:\n1. Add New Movie Genre\n2. Display All Movie
        \"3. Add New Movie to library\n4. Display Movie Library");
        menuChoice = Convert.ToInt32(Console.ReadLine());

        switch (menuChoice)
        {
            case 1:
                try
                {
                    movieGenre = new MovieGenre();
                    Console.WriteLine("Enter Genre Id:");
                    movieGenre.MovieGenreId = int.Parse(Console.ReadLine());
                    Console.WriteLine("Enter movie category:");
                    movieGenre.MovieCategory = Console.ReadLine();
                    movieRepository = new MovieRepository();
                    MovieGenre newGenre =
                    movieRepository.AddNewGenre(movieGenre);
                    Console.WriteLine("New movie Genre added successfully");
                }
                catch
                {
                    Console.WriteLine("New movie Genre addition failed. Check
                    your entry or application logic.");
                }
                break;

            case 2:
                movieRepository = new MovieRepository();
                Console.WriteLine("List of movie categories:");
                Console.WriteLine("{0,-20}{1}", "Genre Id", "Genre Type");
                IList<MovieGenre> GenreList =
                movieRepository.DisplayMovieGenres();
                foreach (var Genre in GenreList)
                {
                    Console.WriteLine("{0,-20}{1}", Genre.MovieGenreId,
                    Genre.MovieCategory);
                }
                break;

            case 3:
                try
                {
                    movieInformation = new MovieInformation();
                    movieRepository = new MovieRepository();
                    Console.WriteLine("Enter movie id:");
                    movieInformation.MovieId =
                    Convert.ToInt32(Console.ReadLine());
                    Console.WriteLine("Choose the movie Genre id:");
                    movieInformation.MovieCategoryId =
                    Convert.ToInt32(Console.ReadLine());
                    Console.WriteLine("Enter the movie name:");
                    movieInformation.MovieName = Console.ReadLine();
                    Console.WriteLine("Enter movie release year:");
                    movieInformation.ReleaseYear =
                    Convert.ToInt32(Console.ReadLine());
                    Console.WriteLine("Enter Viewer Rating(between 1 to 5):");
                    int rating = Convert.ToInt32(Console.ReadLine());
                    MovieFeedback feedback = new MovieFeedback();
                    movieInformation.AudienceFeedback =
                    feedback.GetFeedbackByRating(rating);

```

```

        MovieInformation newMovie =
movieRepository.AddNewMovie(movieInformation);
        Console.WriteLine("New movie added to database");
    }
    catch
    {
        Console.WriteLine("New movie addition failed.");
        Console.WriteLine("Either same movie id already exists in
database or wrong data input.");
    }
    break;

    case 4:
        Console.WriteLine("Display Movie Library:");
        Console.WriteLine("{0,-20}{1,-20}{2,-20}{3,-20}{4}",
            "Movie Id", "Movie Name", "Category", "Release Year",
"Audiance Feedback");
        movieRepository = new MovieRepository();
        IList<MovieInformation> movieList =
movieRepository.DisplayMovieLibrary();
        if (movieList.Count > 0)
        {
            foreach (var movie in movieList)
            {
                Console.WriteLine("{0,-20}{1,-20}{2,-20}{3,-20}{4}",
                    movie.MovieId, movie.MovieName,
movie.Genre.MovieCategory,
                    movie.ReleaseYear, movie.AudienceFeedback);
            }
        }
        else
        {
            Console.WriteLine("No movie is present in library");
        }

        break;

    default:
        Console.WriteLine("Invalid menu entry. Enter 1 or 2 or 3");
        break;

    }
    Console.WriteLine("Enter yes to continue..Any other key to terminate:");
    loopInput = Console.ReadLine();
}
while (loopInput.Equals("yes",
StringComparison.InvariantCultureIgnoreCase));

    Console.WriteLine("Thank you for using the movie application. Have a nice
day");
}
}
}

```