

About DeltaX

DeltaX is an advertising technology startup. We are building a cloud based platform for advertising agencies that integrates with over 10 advertising solution through APIs. Advertisers worldwide can efficiently buy, track, attribute, optimize and report metrics across all media channels (Search, Social, Display RTB, Mobile, Video) using our revolutionary platform.

Do you thrive on challenges? Are you smart, focused and proactive? At DeltaX our goal is to bring together the brightest minds to tackle big engineering and business challenges. If you want to develop state-of-the art technology, we want to hear from you.

Problem Statement

Create a MVC 4 app using Entity Framework Code First to build IMDB.com like website with basic CRUD and movie listing. We would like to have the following entities in the application.

Actors	Movies	Producers
- Name - Sex - DOB - Bio	- Name - Year of release - Plot - Poster (image)	- Name - Sex - DOB - Bio

Relationships

- Actor can act in multiple movies
- Movie can have multiple actors
- Movie has only one producer
- Producer can produce multiple movies

Application specifics

1. Screen to list all movies with Name, Year of release, Producer and all Actors of that movie
2. Screen to 'add' a new movie with the necessary fields with existing actors and producers. If the user wants to add new 'Actors' and 'Producers' while creating the movie which are not present in the database then he should be able to do so while being on the same screen.
3. 'Listing' screen should allow user to click on a movie and view all the details of the movie.
4. 'Listing' screen should allow user to click on 'Edit' and edit all the details of the movie and save it.
5. 'Listing' screen should allow user to click on 'Delete'; pop-up a confirm message (Yes/No) for the user and if the user clicks 'Yes' delete the movie from the Db.
6. Adding / Editing / Deleting a movie should take the user back to the 'Listing' screen with a notification message.

Hints and Tips

1. We are always on the lookout for best practices. Design/Styling/CSS is not critical; functionality is.
2. Use of jQuery / Ajax etc. would be appreciated where applicable.
3. Focus on database design; we are fanatic about Normal Forms.
4. Define your own rules for validation; use common sense - also please make a note of the validations that you have added
5. Surprise us!