# Advanced JavaScript Practical Exam: IMDb

The goal of this sample exam is to practice the creation of a JavaScript structure, using the best practices of **quality JavaScript**, without any help of external libraries such as jQuery.

### The IMDb System

**The IMDb system** is a movies library. IMDb system should have Genres, Movies, Theatres, Actors and Reviews.

### The Source Code

You are given the following components:

* **index.html** - The HTML file of the test page
* **bootstrap.css, style.css** - The styles of application;
* **generator.js** – Generates test data;
* **extensions.js** – Empty file, used to add extension functions (not necessary);
* **html-loader.js** – Loads the categories, users and attaches event handlers;
* **app.js** – app.js is the start point of the application.

Your task is to implement only the missing **models** and extend the **html-loader.js** to properly insert the models in the DOM. If you implement them correctly, your application will run without errors. You **do not need** to edit the **html/css, app.js** and **generator.js** files.

## Implement function constructors or objects for all types

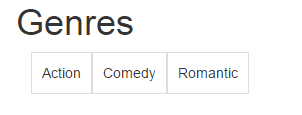
You can use Pseudo-classical or Prototypal OOP.

* Genre – Genre should have name and collection of movies. Movies should be hidden.   
  Add methods:
  + addMovie – Adds movie in movies collection
  + deleteMovie – Deletes movie from movies collection
  + deleteMovieById – Deletes movie from movies collection by given id
  + getMovies – Returns movies collection
* Movie – Movie should have title, length(in minutes)**,** rating(from 1 to 10) and country.Movie should have a collection of actorsand a collection ofreviews. The collections should be hidden.   
  Add methods:
  + addActor – Adds actor in actors collection
  + getActors – Returns actors collection
  + addReview – Adds review in reviews collection
  + deleteReview – Deletes review from reviews collection
  + deteleReviewById – Deletes review from reviews collection by given id
  + getReviews – Returns reviews collection
* Theatre – Theatre should have name, length (in minutes), rating (from 1 to 10), country and isPuppet.
* Actor – Actor should have name, bio (biography) and born (date of birth).
* Review – Review should have author (name of the author), content and date (creation date).
* **Add ids in all objects.** The id should be hidden and should increase on every created instance.
  + **For example:** first instance should have id = 1, second instance should have id = 2 and etc.

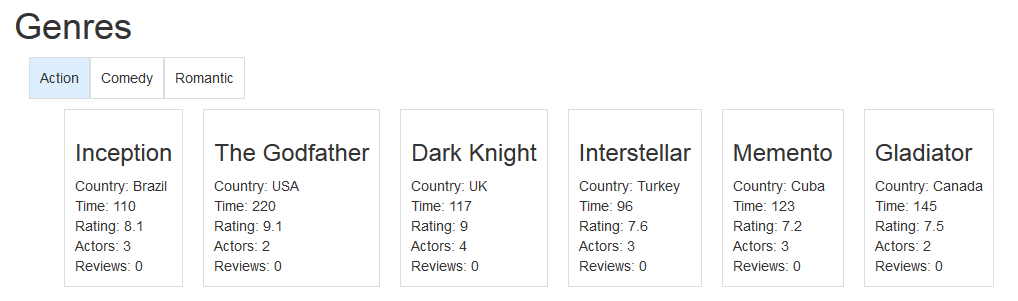
Separate all models in different files. Expose functions for creating instances of current model in the global object “**imdb**”.

You can see how the objects are created in the generator.js file for example.

If you implement correctly all types, the code will work and will show something like this in the browser:



When you click on some genre, you will see all movies in the genre with information about them:



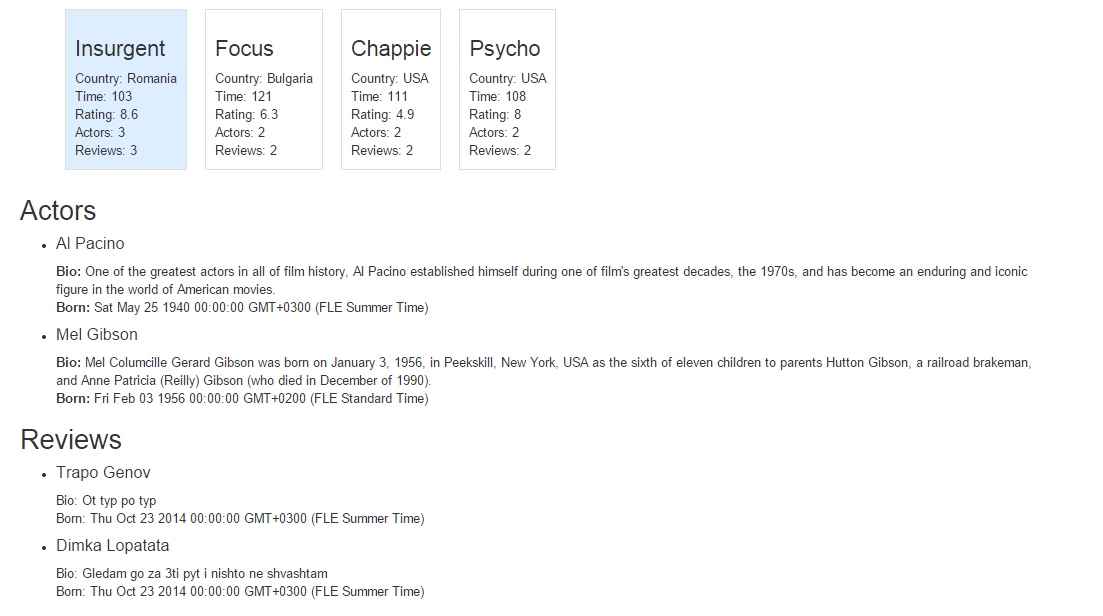
50 points

## Display reviews and actors when click on movie box

Your task is to display more information about the movies when you click on the movie box.

**Tips:**

* You should attach an event handler to the movies container. When the list item inside the container is clicked, the code should be executed.
* You should get the movie data from the data collection. You can get all movies from all genres and then you can filter by id.
* Load movie details in the details container.
  + Load actors with all information about them.
  + Load reviews with all information about them.



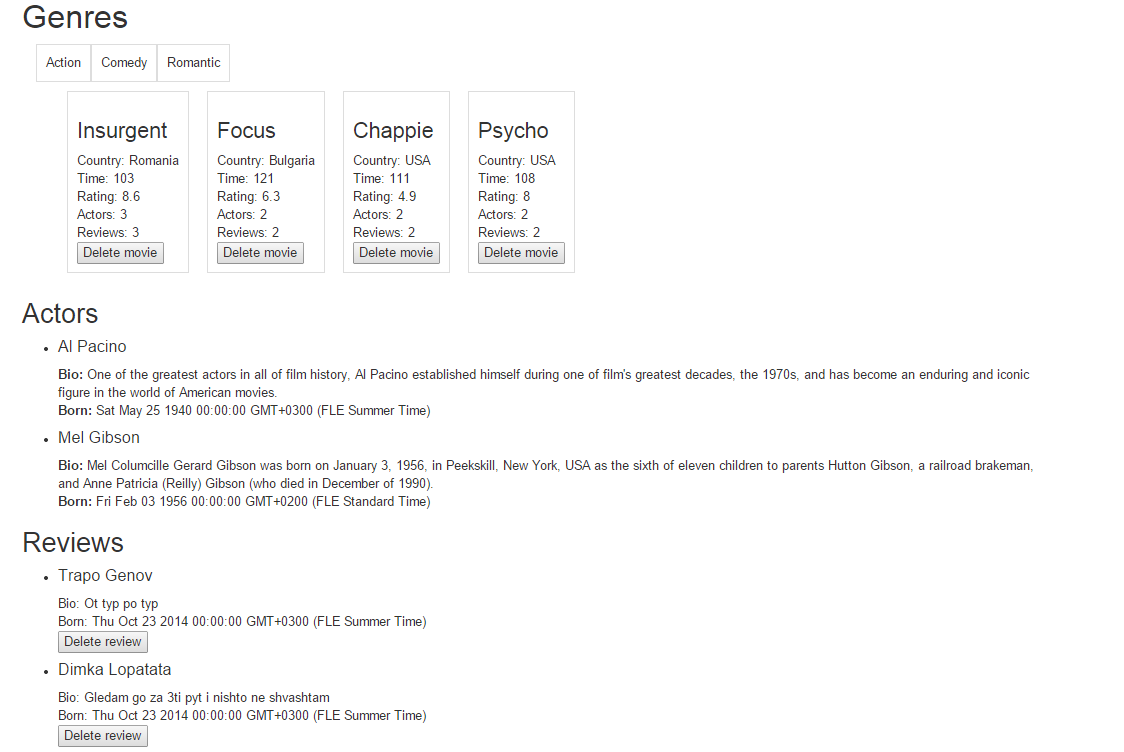
35 points

## Add delete button for movies or reviews

Your task is to generate **either** delete button in the movie box **or** delete button in the review box. When you click on the delete button, the movie or the review should be deleted from the data collection and from the **DOM**.

**Tips:**

* You should attach an event handler on the movies container. When the delete button is clicked, the code should be executed.
* You can get the movie id from the attribute of the list item.
* You can get the genre id from the attribute of the movies container.
* You can get genre from the data collection.
* You can delete movie by id.
* You can delete movie list item from the DOM.



15 points