

Homework Assignment 8: Kevin Bacon

EXTRA FEATURES

The following are **extra features** you can optionally complete. The first extra feature you complete is worth **+1 “oops” point** on HW8; this can be used to make up for a -1 you might receive on some other part of the program, but it cannot earn you more than 100% on HW8, and it cannot be applied to any other programs besides HW8. If you did not lose any points on HW8, your first extra feature is worth **+1 late day** instead. If you choose to complete more than one extra feature, each additional feature you complete is worth **+1 late day**.

NOTE: HW8 is the last assignment, and cannot be submitted past the last day of class. As such, a maximum of 2 late days can be used on it.

If you choose to do any extra features, in your program files' comment headers, indicate which one(s) you did. Regardless of how many additions you write, the page behavior and appearance **should still work as specified**. If you have a different idea for an addition to the program, please ask us and we may approve it.

Extra Feature #1. Select actor from list

By default the **actorid.php** web service returns only the single best match for the given search terms. If you want to return *all* the actors that matched the given name rather than just the first, you can pass an additional query parameter to the service:

parameter name	value
all	when set to any “truthy” value, will cause the service to return all possible matches for the given actor name, rather than just the first

For example, searching (against the full imdb database) for *all* actors matching “ad” and “pitt” will produce:

https://webster.cs.washington.edu/cse190m/homework/hw8/actorid.php?first_name=ad&last_name=pitt&imdb=true&all=true

```
<actors count="3">
  <actor id="376249" first_name="Brad" last_name="Pitt" />
  <actor id="755520" first_name="Adelaide" last_name="Pittaluga" />
  <actor id="782869" first_name="Nadia" last_name="Scarpitta" />
</actors>
```

If there is only one matching actor, the actors tag will be omitted and only the single match will be returned, as if the all parameter were not set.

For this extra feature, modify your search form to display a list of choices whenever multiple actors are returned by **actorid.php**, to let the user choose which actor they meant. The exact user interface for this is left for you to design: Should you display a list of links? a dropdown box? Should the text fields be hidden? etc.

Extra Feature #2. Ajax-ian **search.php** results

Change **search.php** into a web service that spits out a partial HTML page containing just the two tables described previously. Contact it using an Ajax request rather than by a normal form submission, and when it sends back its HTML response, inject this into the main **mymdb.php** page. If you do this feature, a few appearance guidelines from the main spec (“same title/links,” “common search form,” etc.) no longer apply to **search.php** and should be ignored. Your **common.php** file can also be empty, if you don't share any functions between the two files.

Extra Feature #3. Actor's career stats

Report information about the actor's career, such as their number of years they've been working in film (difference between first and most recent film), their average number of years between films, the percentage of their films in which they co-star with Kevin Bacon, their total number of appearances, and so on.

Extra Feature #4. Extended movie information

In addition to the movie info listed previously, also show the genre(s) and/or director name of each movie in which the actor appeared. You'll have to modify your SQL queries and the resulting tables to accommodate this information. Note that a movie can have more than one genre/director; it is important that only a **single row** is displayed in the table for every such movie, despite the (possible) presence of multiple genres/directors in your results.

Extra Feature #5. Bling!

Add several Scriptaculous effects to your page to make it more awesome.

Extra Feature #6. Parameterize both actors

The normal assignment lets the user enter one actor to search for, but the other actor (Kevin Bacon) is fixed. For this feature, make it so that both actors are provided by the user. The user types in two first/last names, and the [search.php](#) page will show three tables instead of two: one table for the films they appeared in together, and separate tables for each of the two actors listing all the movies in their career.

If you do this feature, it's okay if each of the two actors requires a separate Ajax request to [actorid.php](#) to get the actor's unique ID. But otherwise you should have the same kinds of queries as described in the main spec. Use parameters in your [search.php](#) code to avoid redundancy between similar queries/code between the two actors.

Extra Feature #7. Web security

Modify your code to be secure against web security threats discussed in lecture. Encode every query parameter accepted by your PHP pages before outputting them in the page, to prevent HTML injection and cross-site scripting. Also encode any values passed to your pages that are to be passed to SQL queries, to prevent SQL injection attacks.

Extra Feature #8. Session/cookie data for last search

Use sessions and/or cookies as discussed in lecture to remember values that the user has entered into the page. For example, remember most recent actor's first/last name so that if the same user returns to the page later, those values will be automatically inserted into the input text boxes.