

**University of Washington, CSE 154**  
**Homework Assignment 6: Kevin Bacon**  
**EXTRA FEATURES / CSE Majors Features**

CSE majors (B section) must complete **at least one (1)** of the following extra features to add new functionality to the program. If a CSE major completes more than one extra feature successfully, they earn +1 extra late day for use on a future assignment (*after HW6*). (Not +1 late day per additional feature completed; +1 overall.)

Non-CSE majors (A section) are not required to complete any of these extra features, but if they do, each optional feature completed is worth +1 late day for use on a future assignment (*after HW6*), up to a maximum of +2.

Regardless of how many additions you write, the page behavior and appearance should still work as described in the main spec document. If you have a different idea for an extra feature, please ask us and we may approve it.

If you choose to do any of these extra features, in your file's comment header, indicate which one(s) you did.

**Extra Feature #1. Select central actor from list**

Normally the site uses Kevin Bacon as the "central" actor, the one compared against all other actors that the user searches for. Modify your site so that each page instead contains a list (such as a `select` box) of available central actors to choose from. The user can pick one from the list and view their own actor's films in common with that central actor.

**Extra Feature #2. 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, and so on.

**Extra Feature #3. 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. You may assume that every film has at least one genre and at least one director (that is, at least one row in each of the `movies_directors` and `movies_genres` tables that match its id). Note that a movie can have more than one genre/director; your PHP code should show a **single row** for each film, containing all information for that film. To achieve this, you can either loop over the rows of your query and examine which ones correspond to the same film and group them together, or you can use a bit of advanced SQL syntax called `GROUP_CONCAT` to concatenate the genres/directors together. See the following link:

[http://dev.mysql.com/doc/refman/5.0/en/group-by-functions.html#function\\_group-concat](http://dev.mysql.com/doc/refman/5.0/en/group-by-functions.html#function_group-concat)

**Extra Feature #4. Validation and security**

Modify your code to be robust against bad input and resistant against various common attacks. Encode every query parameter accepted by your PHP pages using the `htmlspecialchars` function before outputting them in the page, to prevent HTML injection and cross-site scripting. Encode every parameter placed into a SQL query using the PDO object's `quote` function as described in the textbook, to prevent SQL injection attacks. For all GET or POST query parameters, check to make sure that the parameters are passed and contain non-empty values; if they are absent or empty, abort the page (`die`) or present a helpful error message.